

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 1**



“Módulo de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1.”

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Richard Alejandro Castañet Blanco

Tutores:

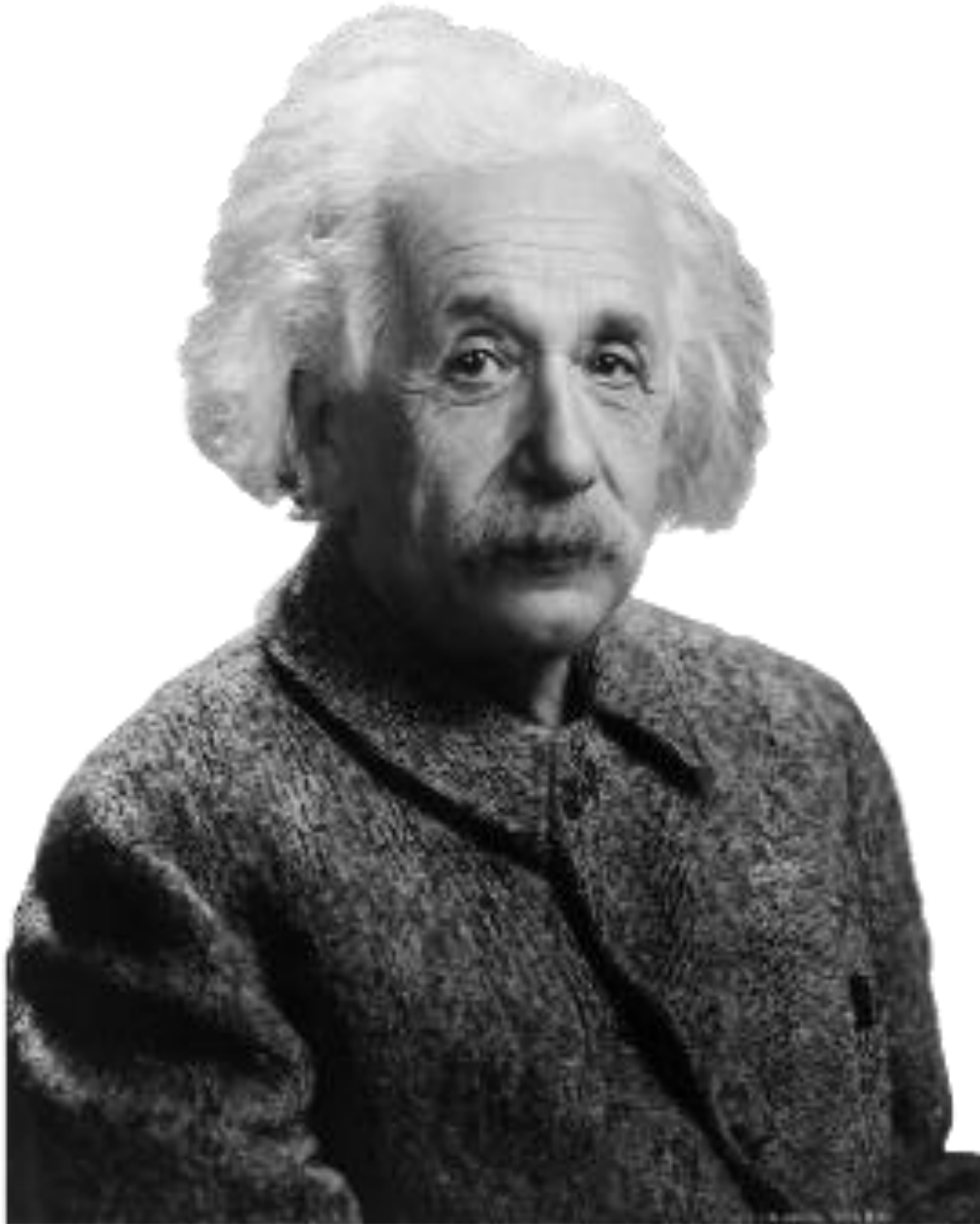
Msc. Disnayle Jorge Chacón

Msc. Yaniel Lázaro Aragón Barreda

LA HABANA, NOVIEMBRE 2022

Año 64 de la Revolución

“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía atómica: la voluntad.”



Albert Einstein

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título "***Módulo de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1***", concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los <día> días del mes de noviembre del año 2022.

Richard Alejandro Castañet Blanco

Firma del Autor

<nombre del tutor>

Firma del Tutor

<nombre del tutor>

Firma del Tutor

Síntesis del tutor o los tutores:

Disnayle Jorge Chacón: Graduada de Ingeniero en Ciencias informáticas en julio del 2010 en la Universidad de las Ciencias Informáticas (UCI), Profesora del departamento de Informática de la facultad 1. Actualmente se desempeña como profesora principal de 5to año. Ha impartido las asignaturas Proyecto de Investigación y Desarrollo IV, V, VI, VII, Introducción a las Ciencias Informáticas, Introducción a las Ciencias Informáticas I y II, Algebra Lineal y Matemática I. Máster en Ciencias Matemáticas: Mención Enseñanza de las Matemáticas en la Universidad de la Habana.

Yaniel Lázaro Aragón Barreda, Profesor Asistente, Máster en Educación a Distancia, Profesor del Departamento de Informática en la Facultad 1. Graduado de Ingeniería en Ciencias Informáticas en el 2013. Se desempeña como Profesor Principal de Año Académico y Profesor de Sistemas de Bases de Datos. Tutor de varios trabajos de diploma en pregrado y miembro de tribunales al efecto. Ha cursado e impartido postgrados relacionados con la especialidad.

AGRADECIMIENTOS

A mi familia, profesores, tutores, y amigos que me han acompañado a lo largo de esta experiencia maravillosa, en especial a mis padres, hermano, mi novia, y amigos cercanos. Gracias a todos.

DEDICATORIA

A mi madre por siempre estar a mi lado en todo momento, a mi padre por su apoyo incondicional, a mi novia, una de las personas más importante de mi vida, y a mis abuelos por sus sabios consejos. A todos esos amigos, que hicieron de la convivencia durante estos 5 años momentos maravillosos, y a todos esos que no están día a día, pero no dejaron de preocuparse. A mi hermano, por su total entrega y amor, y esos tutores y profesores que ayudaron a que hoy se obtuviera este resultado.

Resumen

En la Facultad 1 de la Universidad de las Ciencias Informáticas, se lleva a cabo la planificación y confección del Horario Docente. Actualmente, se están presentando ciertas dificultades con dicho proceso docente-educativo, debido a que es un proceso que se realiza de forma manual, lo que ocasiona que se puedan cometer diversos errores humanos por parte de la persona encargada de realizarlo. La investigación en curso propone un módulo de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1, capaz de generar la planificación y confección de manera automática del Horario Docente de la Facultad. Para ello se estableció el estado del arte asociado a varios módulos y sistemas similares, que fundamentan las funcionalidades del mismo. Además, se hará uso de la Metodología AUP, en su variante realizada para la Universidad de las Ciencias Informáticas, así como el uso de varias herramientas y tecnologías entre las que se encuentran: lenguaje de programación PHP, Framework Laravel, Visual Paradigm, Gestor de Base de Datos Xampp, entre otros. También, se planificó y diseñó una estrategia de pruebas con el objetivo de validar las principales funcionales del módulo, así como la seguridad del mismo, entre otros detalles. Con el desarrollo de este módulo se espera obtener una herramienta de trabajo capaz de lograr realizar una rápida planificación y control del horario docente de la facultad 1.

Palabras Claves: confección, horario docente, módulo, planificación

Abstract

In Faculty 1 of the University of Informatics Sciences, the planning and preparation of the Teaching Schedule is carried out. Currently, there are certain difficulties with said teaching-educational process, because it is a process that is carried out manually, which causes various human errors to be made by the person in charge of carrying out the process. The ongoing research proposes a module for the generation of teaching schedules for the Educational Work System in Faculty 1, capable of automatically generating the planning and preparation of the Teaching Schedule of Faculty 1. For this, the associated state of the art was established. to various modules and similar systems, which support its functionalities. In addition, the AUP Methodology will be used, in its variant made for the UCI, as well as the use of various tools and technologies among which are: PHP programming language, Laravel Framework, Visual Paradigm, Xampp Database Manager, among others. Also, a testing strategy was planned and designed with the objective of validating the main functionalities of the module, as well as its security, among other details. With the development of

this module, it is expected to obtain a work tool capable of achieving rapid planning and control of the teaching schedule of the faculty 1.

Key Words: *planning, preparation, module, teaching schedule.*

Índice

INTRODUCCIÓN	3
CAPÍTULO 1: FUNDAMENTOS Y REFERENTES TEÓRICOS-METODOLÓGICOS SOBRE EL PROCESO DE CONFECCIÓN DEL HORARIO DOCENTE DE LA FACULTAD 1 PARA EL SISTEMA DE TRABAJO EDUCATIVO EN LA FACULTAD 1	8
1.1 Conceptos asociados a la planificación y confección del horario docente.	8
1.2 Análisis comparativo de Módulos y Sistemas de Generación de Horarios Docentes	9
1.2.1 Sistemas y Módulos de Gestión de Horarios Internacionales	9
1.2.2 Sistemas y Módulos de Gestión de Horarios Nacionales	13
1.3 Metodología, herramientas y tecnologías para dar solución al problema	15
CONCLUSIONES PARCIALES DEL CAPÍTULO	23
CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL MÓDULO DE GENERACIÓN DEL HORARIO DOCENTE PARA EL SISTEMA DE TRABAJO EDUCATIVO EN LA FACULTAD 1	24
2.1 Descripción actual del proceso de planificación de Horarios Docentes en la Facultad 1. ...	24
2.2 Descripción de la propuesta de solución para la generación de horarios docentes.	26
2.3 Análisis de la propuesta de solución para la confección del horario docente.	26
2.3.1 Técnicas de Captura de Requisitos	27
2.3.2 Requisitos Funcionales	27
2.3.3 Requisitos No Funcionales	32
2.3.4 Diagrama de Caso de Uso del Sistema	34
2.3.5 Descripción de Casos de Uso del Sistema	35
2.4 Diseño de la propuesta de solución para la confección del horario docente.	47
2.4.1 Diagramas de clases de diseño con estereotipos web	47
2.4.2 Diagramas de secuencias	51
2.4.3 Modelo de datos	55
2.4.4 Arquitectura	57
2.4.5 Patrones de diseño	58
2.4.6 Diagrama de despliegue	61
CONCLUSIONES PARCIALES DEL CAPÍTULO	63

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO DE GENERACIÓN DE HORARIOS DOCENTES PARA EL SISTEMA DE TRABAJO EDUCATIVO EN LA FACULTAD 1.	64
3.1 Introducción	64
3.2 Diagrama de Componentes	64
3.3 Estándares de codificación	70
3.4 Estrategia de Pruebas	72
3.4.1 Pruebas Funcionales	73
3.4.2 Pruebas de Seguridad	79
3.4.3 Pruebas de Integración	81
3.4.4 Pruebas de Carga y estrés	83
3.5 Resultados del MGHDF1	84
.....	88
3.6 Validación Científica	89
CONCLUSIONES PARCIALES DEL CAPÍTULO	90
CONCLUSIONES GENERALES	92

Índice de Tablas

Tabla 1.1 Resumen comparativo de los Sistemas y Módulos de Gestión de Horarios Internacionales.	12
Tabla 1.2 Resumen comparativo de los Sistemas y Módulos de Gestión de Horarios Nacionales.	15
Tabla 2. Requisitos Funcionales	27
Tabla 3. Descripción del Diagrama de Componentes Gestionar Asignaturas.....	68
Tabla 3.1. Descripción del Diagrama de Componentes Gestionar Afectaciones.	68
Tabla 3.2. Descripción del Diagrama de Componentes Gestionar Balance de Carga.	69
Tabla 3.3. Estándares de Codificación.....	70
Tabla 3.4. Estrategias de Pruebas MGHDF1.	72
Tabla 3.5. Pruebas Funcionales (Insertar Asignaturas).....	73
Tabla 3.6. Caso de Prueba 1: RF6_Insertar Asignatura.	74
Tabla 3.7. Pruebas Funcionales (Insertar Balance de Carga).	74
Tabla 3.8. Caso de Prueba 2: RF17_Insertar Balance de Carga.....	75
Tabla 3.9. Pruebas Funcionales (Insertar Afectaciones).	76
Tabla 3.10. Caso de Prueba 3: RF11_Insertar Afectaciones.....	77
Tabla 3.11. Pruebas de Cargas y Estrés.....	83
Tabla 3.12. Cuadro lógico de ladov (Elaboración Propia)	89
Tabla 3.13. Resultados Obtenidos de la Encuesta.....	90

Índice de Figuras

Figura 1. Modelo Conceptual	25
Figura 2 Diagrama de Casos de Usos del Sistema.....	35
Figura 3. Diagrama de Clases con Estereotipos Web (Gestionar Asignaturas).....	48
Figura 4. Diagrama de Clases con Estereotipos Web (Gestionar Balance de Carga).....	49
Figura 5. Diagrama de Clases con Estereotipos Web (Gestionar Afectaciones).....	50
Figura 6. Diagrama de Secuencia (Insertar Asignatura).....	52
Figura 7. Diagrama de Secuencia (Editar Balance de Carga)	53
Figura 8. Diagrama de Secuencia (Eliminar Afectaciones).....	54
Figura 9. Modelo de Datos.....	56
Figura 10. Modelo-Vista-Controlador	57
Figura 11. Empleo del Patrón Experto	59
Figura 12. Empleo del Patrón Alta Cohesión.....	60
Figura 13. Uso del patrón de diseño GOF (Factory).....	61
Figura 14. Uso del Patrón Decorador.....	61
Figura 15. Diagrama de Despliegue.....	62
Figura 16. Diagrama de Componentes Gestionar Asignaturas.....	65
Figura 17. Diagrama de Componentes Gestionar Afectaciones.....	66
Figura 18. Diagrama de Componentes Gestionar Balance de Carga.....	67
Figura 19. Pruebas de Funcionalidad.....	79
Figura 20. Herramienta Acunetix.....	80
Figura 21. Pruebas de Seguridad	81
Figura 22. Pruebas de Integración	82
Figura 23. Pruebas de Integración-Github	82
Figura 24. Portada Inicial del Sistema de Trabajo Educativo de la Facultad 1.	85
Figura 25. Módulo Horario.	86
Figura 26. Vista Generar-Horario	87
Figura 27. Visualizar Horario.....	88

INTRODUCCIÓN

El constante desarrollo de las Tecnologías de la Información y las Comunicaciones, conocidas por sus siglas ¹TIC, ha obligado a un incremento de su uso a lo largo del mundo en innumerables sectores del desarrollo, tanto educativo, como político, como comercial, como social, entre muchos más. Hoy en día es imprescindible el empleo de las TIC en casi todos los sistemas educativos del mundo actual, debido al avance de las tecnologías y su tan frecuente y ascendente uso diario por cada profesor y estudiante en general. Es válido destacar que su empleo supone un significativo avance en los sistemas educacionales de todos los países, incluyendo Cuba, donde sin lugar a dudas ha marcado un antes y un después en lo que a la cultura informática y educativa se refiere, quedando demostrado esto en los logros obtenidos en la mayoría de las instituciones educativas del país.

El entorno educativo actual a nivel mundial se caracteriza por acelerados cambios docentes-educativos, los cuales, fortalecido por las innovaciones continuas en el campo de las tecnologías informáticas, han originado un despliegue multidireccional en las actividades docentes basadas en las comunicaciones. Las comunidades humanas han ampliado sus horizontes geográficos para disponer de la información con una facilidad nunca antes vista. La información y el conocimiento se han convertido en valores de uso muy importantes de la sociedad global (Rodríguez, 2017).

Es imprescindible la creación de condiciones necesarias y suficientes que permitan explotar al máximo todas las ventajas que nos ofrecen el avance desenfrenado de las tecnologías, contando siempre con el correcto y racionado uso de los medios, canales e infraestructura creados por cada sistema educativo para lograr un alto nivel de su uso y por ende poder entregar a los estudiantes y personas interesadas una educación a la altura del nivel institucional y educacional de Cuba.

La Universidad de las Ciencias Informáticas (UCI) no se encuentra para nada ajena al uso y disfrute de las Tecnologías de la Información y las Comunicaciones, ha sido todo lo contrario, pues estas han constituido un fuerte apoyo al desarrollo del proceso docente-educativo en la Universidad. A pesar del empeño puesto en el uso de las TIC en la Universidad, aún se ponen de manifiestos varios problemas a través del uso y disfrute de las mismas, siendo un ejemplo de esto es la planificación del Horario Docente de la Facultad 1.

¹ Las TIC pueden definirse como las tecnologías que utilizan la informática, la microelectrónica y las telecomunicaciones para crear nuevas formas de comunicación con el fin de facilitar la emisión, acceso y tratamiento de la información.

Un Horario docente constituye el instrumento organizativo de las actividades docentes, que permite brindar una visión general de la distribución cuantitativa de las asignaturas contenidas en el plan de estudio, según el número de horas de clases cada año.

La planificación de un horario docente es un proceso engorroso para el cual se han usado varios métodos (coloreo de grafos, recocido simulado, algoritmos genéticos, entre otros), que aplicados en un software son capaces de resolver el problema de planificación de horarios docentes, aunque ninguno ha logrado realizar una planificación que se adapte a todo tipo de institución.

Para confeccionar el horario docente se deben tener en cuenta varios aspectos: la asignación de locales destinados a la docencia, el número de profesores por año, la cantidad de asignaturas por año, las horas destinadas a la docencia y a la producción, la cantidad de grupos por año, la planificación de las asignaturas, el balance de carga docente del año, las afectaciones de los profesores del año, entre otras actividades. Se han desarrollado software a nivel mundial que ayudan a la elaboración de los horarios docentes en centros educacionales, pero ninguno cumple con las exigencias de la universidad porque la mayoría han sido llevados a cabo para centros pequeños y de poca matrícula.

En la Universidad existen aplicaciones desarrolladas que facilita la elaboración del horario docente, pero no cuenta con todas las funcionalidades necesarias para lograr un óptimo resultado. Lo que provoca que el trabajo del Vicedecano de Formación, la planificadora y los profesores principales se vea limitado en cuanto a la eficiencia, ocasionando insatisfacción en aquellos que puedan estar involucrados, molestias innecesarias y pérdida de tiempo en ocasiones.

Otro tema de importancia son los locales destinados a las labores estudiantiles que tiene cada facultad y el elevado número de estudiantes, resulta difícil lograr una buena organización teniendo en cuenta la concurrencia de grupos y profesores que deben impartir clases a más de 2 grupos, provocando que el trabajo sea engorroso, pues una vez elaborado el horario tiene que corregirse.

Por todo lo anteriormente planteado y argumentado se plantea como **Problema de Investigación:** ¿Cómo contribuir a la generación del horario docente desde el Sistema de trabajo educativo de la Facultad 1?

La presente investigación está marcada por el **objeto de estudio:** Proceso de planificación de las actividades docentes, haciendo énfasis en el **campo de acción:** Proceso de confección del Horario Docente de la Facultad 1.

Teniendo en cuenta el problema identificado y el objeto de estudio, se propone como **objetivo general:** Desarrollar el módulo de generación de horarios docentes para el sistema de gestión del Trabajo Educativo en la Facultad 1.

Entre las **preguntas científicas** que nos podemos encontrar están:

1. ¿Cuáles son los referentes teóricos fundamentales que sustentan la investigación relacionada con el proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1?
2. ¿Qué propuesta de solución se define para el proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1?
3. ¿Cuáles son las características que debe cumplir el proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1?
4. ¿Cómo validar el funcionamiento del proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1?

Para responder las anteriores preguntas científicas es necesario dar cumplimiento a las siguientes **tareas de investigación:**

1. Estudio de los referentes teóricos cruciales que sustentan la investigación relacionada con el proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1.
2. Análisis y diseño del módulo para la gestión del proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1.
3. Implementación de las funcionalidades del módulo para el proceso de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1.
4. Validación de las funcionalidades del módulo para la generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1.

Métodos científicos de investigación:

El método científico es el medio para conocer y comprender las realidades y procesos del mundo que nos rodea mediante un estudio ordenado, sistemático y sistémico. Si no hay método científico, no hay ciencia, de hecho, la principal diferencia que existe entre el conocimiento científico y el conocimiento ordinario es que el primero se alcanza después de una investigación y análisis consciente de lo ya conocido.

En la investigación que se está llevando a cabo se asumen para su correcto desarrollo los siguientes métodos de investigación:

Entre los **métodos teóricos** utilizados están:

1. **Modelación:** Se modela el desarrollo de la solución a partir de los artefactos que establece la metodología de desarrollo y el modelo de datos de la solución propuesta.
2. **Análisis Sintético:** Permite concretar y resumir el conocimiento obtenido de los materiales estudiados sobre la planificación y confección de horarios docentes, así como la descripción y análisis de los procesos observados con el fin de emplearlo en el desarrollo de la presente investigación.

Los **métodos empíricos** ayudan a complementar el seguimiento a la evolución del proceso que se estudia, los utilizados en esta investigación son:

1. **Entrevista:** Realizadas a los implicados en el proceso de planificación y control docente de la Facultad 1, que permite recopilar datos acerca del estado que presenta actualmente el proceso de planificación y confección del horario docente, durante varias reuniones informales realizadas en varios lugares de la Facultad 1 (Ver Anexo 1).
2. **Observación:** Facilita obtener conocimiento acerca del funcionamiento de los sistemas y módulos existentes para la planificación y confección de módulos de gestión de horarios docentes.

Para la presente investigación se empleará como **método de validación** el siguiente:

1. **Técnica de ladov:** La técnica de ladov constituye una vía indirecta para el estudio de la satisfacción, debido a que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas que se intercalan dentro de un cuestionario aplicado y cuya relación el sujeto desconoce (Castro, 2020). Para la aplicación de la técnica se realizó una encuesta a los cinco Profesores Principales de cada año, con el objetivo de conocer el Índice de Satisfacción Grupal de los mismos con respecto al módulo desarrollado, dicha encuesta se puede observar en el Anexo 3.

El presente trabajo está estructurado de la manera siguiente:

Capítulo 1: Fundamentos y referentes teóricos-metodológicos sobre el proceso de confección del horario docente de la Facultad 1 para el Sistema de Trabajo Educativo en la

Facultad 1: Este capítulo aborda las cuestiones teóricas necesarias para el análisis y comprensión de la investigación, la cual tiene presente el establecimiento del estado del arte mediante el estudio de herramientas de planificación y confección tanto a nivel internacional como nacional. Se lleva a cabo un amplio análisis, estudio y selección de la metodología de desarrollo en función de los requerimientos del sistema y las características del equipo de trabajo, así como el entendimiento de las herramientas y tecnologías a utilizar.

Capítulo 2: Análisis y diseño del módulo de planificación y confección del horario docente de la facultad 1 para el Sistema de Trabajo Educativo en la Facultad 1:

En este capítulo se realiza una posible solución evidenciándose los principales requisitos y funcionalidades del módulo para la planificación y confección del Horario Docente de la Facultad 1. Se pone de manifiesto todo el diseño del módulo a través de los artefactos definidos por la metodología para esta fase.

Capítulo 3: Implementación y validación del módulo de generación de horarios docentes para el Sistema de Trabajo Educativo en la Facultad 1:

En este capítulo se elabora la posible propuesta de solución, poniéndose de manifiesto las funcionalidades del módulo para la planificación y confección del Horario Docente de la Facultad 1. Se muestran los resultados de la validación de los requisitos y diseños, así como los resultados obtenidos de la validación científica realizada. Además, se lleva a cabo un resumen de los resultados obtenidos tras la aplicación de las distintas pruebas de funcionamiento efectuadas al módulo.

CAPÍTULO 1: FUNDAMENTOS Y REFERENTES TEÓRICOS-METODOLÓGICOS SOBRE EL PROCESO DE CONFECCIÓN DEL HORARIO DOCENTE DE LA FACULTAD 1 PARA EL SISTEMA DE TRABAJO EDUCATIVO EN LA FACULTAD 1

En el presente capítulo se estudian y analizan los conceptos fundamentales relacionados a los módulos de confección de horarios docentes con el objetivo de dar soporte teórico a la investigación. Además, se realiza un detallado estudio y análisis de módulos de generación de horarios docentes a nivel internacional y nacional. De igual manera, se destacan las principales características de la metodología utilizada en la investigación, así como de las tecnologías y herramientas a emplear en la construcción de la propuesta de solución.

1.1 Conceptos asociados a la planificación y confección del horario docente.

A continuación, se presentan conceptos asociados con la planificación y confección de horarios docentes. Recaltar entre los conceptos principales: planificación, actividad docente, control de la actividad docente, pues son relevantes para el desempeño de la investigación con la calidad requerida.

Planificación

La planificación consiste en definir las metas de la organización, establecer una estrategia general para alcanzarlas y trazar planes exhaustivos para integrar y coordinar el trabajo de la organización. Se ocupa generalmente tanto de los fines como de los medios, y de la manera que hay que hacerlo (González, 2018).

La planeación se puede definir como el proceso que busca establecer y elegir objetivos y medios adecuados previos a realizar la acción. Además, se puede afirmar que la planeación tiene la función de establecer el curso, principios, secuencias y tiempos para efectuar la acción (Flores, 2015).

En la presente investigación se asume **planificación** como un proceso bien meditado y con una ejecución estructurada, con el fin el obtener uno, o varios objetivos determinados, de forma que se satisfagan todas las necesidades por las cuales se está ejecutando la misma.

Actividad Docente

Se puede definir Actividad Docente como el conjunto de acciones llevadas a cabo y planificadas tanto por docentes como estudiantes, que tiene como objetivo principal lograr la mayor cantidad de objetivos y finalidades de la enseñanza (Rojas, 2015).

La actividad docente puede ser además la forma básica de la organización del proceso de enseñanza o el arreglo didáctico de la ciencia que transcurre de tarea en tarea, por ello su estructuración adecuada constituye una etapa fundamental del trabajo del educador, para lograr el único objetivo esencial de su labor como docente ya sea de la enseñanza que sea, dígame universitario, primaria, secundaria, entre otros (León, 2022).

A los efectos de la presente investigación se pueden considerar como **actividades docentes** las formas de organización que establecen el reglamento docente de la facultad: Conferencia, Laboratorio, Clase Práctica, Prueba Parcial, Taller, Seminario, Corte de Tarea Extraclase, Corte de Trabajo de Curso, Defensa de Trabajo Extraclase, Defensa de Trabajo de Curso y Pruebas Finales.

Control de la Actividad Docente

El control de la actividad docente es aquel que se realiza a una de las formas organizativas del proceso docente educativo, previsto en el horario de clases de los estudiantes, y estará dirigido a comprobar el logro de los objetivos propuestos para dicha actividad. Al finalizar el control, sin la presencia de los estudiantes, el responsable dirigirá el análisis, dará las conclusiones al controlador, señalará los principales logros, las deficiencias y las recomendaciones. El control de la actividad docente se calificará utilizando las categorías de Excelente, Bien, Regular o Mal. Se exceptuarán de una calificación, los casos que se determinen por el responsable del control (artanaisi, 2019).

Es importante asumir para la presente investigación que se cumple estrictamente con lo establecido en las pautas del control de las actividades docentes de la Facultad, de modo que se respete el trabajo realizado por la persona encargada de la planificación y confección del horario docente de la Facultad 1.

1.2 Análisis comparativo de Módulos y Sistemas de Generación de Horarios Docentes

1.2.1 Sistemas y Módulos de Gestión de Horarios Internacionales

Actualmente existen diversos módulos de gestión de planificación y confección de horarios docentes, teniendo la mayoría en común que cuentan con funcionalidades para la distribución, planificación, confección, y edición de horario docente. A nivel global, entre los más relevantes se encuentran:

FET, el generador de horarios libre

Es una aplicación de software libre y de código abierto para generar horarios académicos para cualquier tipo de institución. Utiliza un algoritmo de programación de horarios rápido y eficiente, y está licenciado bajo GNU GPL. Por lo general, FET es capaz de resolver un calendario complicado en un máximo de 5 a 20 minutos. Para realizar los horarios más simples, puede tardar un tiempo más corto, menos de 5 minutos. Evidentemente, para construir los horarios extremadamente difíciles, puede tardar un tiempo más largo, como horas. Una vez introducidos todos los datos en la aplicación basta con pedir que genere el horario, el cual se puede consultar dentro del mismo programa desde diferentes vistas: por profesor, por clase, por materia, etc. Al generarlo nos informará de los problemas que tiene para su confección, o las condiciones que ha roto para hacerlo (baltolkien, 2014).

Además, genera un documento en HTML, perfecto para colgar en la red, con todas las posibles vistas, y compartir con el resto de usuarios del horario. Y no solo eso, el programa puede crear múltiples horarios para que se pueda elegir cual es la mejor opción para el centro.

Entre las principales desventajas que presenta este software se encuentran:

- I. Debe adquirirse una licencia de pago para su uso.
- II. Debe ser usado bajo el sistema GNU.

Untis

Programa líder mundial en la realización automática de horarios para colegios y universidades. Su potente algoritmo permite configurar un horario totalmente optimizado, según los criterios pedagógicos introducidos y aprovechando al máximo todos los recursos y espacios. Está presente en más de 80 países y disponible en más de 35 idiomas. Además, cuenta con 25.000 centros de usuarios entre los que se encuentran: centros educativos y de formación, escuelas de negocio y universidades (Educaria, 2022).

Entre las principales ventajas y desventajas de estos tipos de horarios flexibles están:

Ventajas:

- I. Se adapta a las necesidades. Con un horario flexible, los empleados pueden organizar su tiempo sin descuidar sus actividades personales, realizando su trabajo cuando sean más productivos o modificando sus horas de entrada y salida.

- II. Mejora la salud mental. Un empleado que puede organizar su propio tiempo tiene una vida más equilibrada. Puede pasar más tiempo ejecutando otras actividades, lo que ayudará a reducir el estrés y aumentará la productividad y creatividad.
- III. Crea un sentimiento de libertad. La flexibilidad de horarios permite a los colaboradores sentirse con el poder de administrar sus tiempos. Esto le ayuda a la empresa a mejorar su imagen, el desempeño de los empleados y los resultados que obtienen.

Desventajas:

- I. Puede afectar la comunicación. Un empleado que trabaja de acuerdo a sus horarios, suele ser más independiente. El contacto con el equipo de trabajo, compañeros y superiores puede verse afectado, por ello es necesario establecer estrategias de comunicación e integración.
- II. La organización puede ser difícil. En todas las empresas los resultados deben ser entregados en tiempo y forma, pero si uno de los empleados falla puede afectar a todos. Si no se logra una buena organización se puede fomentar la irresponsabilidad, la indisciplina y malos resultados.
- III. Los tiempos de trabajo pueden extenderse. Al tener flexibilidad de horarios, pueden existir malos entendidos y los empleados se sentirán obligados a atender peticiones a cualquier hora. No establecer normas puede generar un sentimiento negativo en el equipo y afectar su rendimiento.

GHC

GHC busca soluciones completas de forma muy eficiente. Si existen, los descubre en cortos periodos de tiempo, incluso en los casos más complicados. Una vez que tenga resultados completos, obtendrá un compromiso óptimo y justo para las preferencias y los criterios de ponderación que haya establecido. Para las escuelas primarias, secundarias, preparatorias, escuelas de formación profesional, colegios, universidades u otras instituciones educativas, GHC resuelve los horarios semanales. Puedes configurar diferentes horarios superpuestos en una misma semana y más de una semana con diferentes horarios. GHC es más útil cuanto más difícil es encontrar una solución válida y óptima para su escuela (Peñalara, 2022).

Las principales ventajas que podemos obtener al utilizar este software son:

- I. El software de horarios de GHC es tan fácil de usar, no es necesario ser un experto en informática para usarlo.
- II. El motor es muy potente; consigue resultados rápidamente y los optimiza.
- III. La optimización de horarios observa tus preferencias y criterios para obtener la mejor solución, la que requiere tu institución educativa.
- IV. Peñalara Software mantiene un compromiso constante con sus usuarios, brindándoles actualizaciones continuas del software.

Horario

El programa Horario genera automáticamente los horarios de clase de los profesores de colegios y universidades, de acuerdo con la cantidad de horas semanales de cada asignatura y a la disponibilidad horaria de cada profesor. Calcula los horarios completos de profesores y grupos en minutos, y es muy fácil de utilizar. Los algoritmos de ordenación son de creación original, creados específicamente y mantenidos por Custombit para la generación de horarios docentes de la más alta calidad. Genera de forma automática los horarios de un colegio con tres turnos, y no tiene límite de profesores, materias o cursos. El programa genera los horarios de las clases con la información de los profesores y sus asignaturas, que se va anotando de una manera muy simple, rápida e intuitiva (Comes, 2022).

El estudio y análisis de los sistemas anteriormente descritos nos llevan a la conclusión de que existen varias deficiencias en los mismos, por lo que no pueden ser utilizados y explotados para la solución deseada por la investigación. Entre algunas de las principales deficiencias se pueden observar que por lo general son software de propietarios, y algunos no presentan una buena flexibilidad. Todos los sistemas estudiados a nivel internacional tienen como principal ventaja que generan de manera automática el Horario Docente, pero incluso así no se pueden utilizar para dar solución al problema planteado por la investigación debido al resto de aspectos negativos que poseen.

Tabla 1.1 Resumen comparativo de los Sistemas y Módulos de Gestión de Horarios Internacionales.

Parámetros	FET	Untis	GHC	Horario
Tipo de Software Libre	Sí	No	No	No
Licencia de Pago	Sí	Sí	Sí	Sí

Generación automática del horario	Sí	Sí	Sí	Sí
Tecnología de desarrollo	IA	-	-	-
Afectaciones docentes	Sí	No	Sí	No
Flexibilidad	Sí	Sí	No	No

1.2.2 Sistemas y Módulos de Gestión de Horarios Nacionales

En lo que al nivel nacional respecta, existen varios módulos y sistemas ya desarrollados con el fin de obtener una buena calidad en la planificación y confección de los horarios docentes de las distintas universidades y entidades del país. Se toma en la presente investigación como referencia a la Universidad de las Ciencias Informáticas, debido a la contribución de la misma a las particularidades de la presente investigación y la planificación docente. Se analizaron varios sistemas para la confección y planificación de horarios docentes, entre los que se encuentra:

Implementación de una Biblioteca de Algoritmos para la generación automática de horarios de la aplicación web Ghordo

En este trabajo de diploma se propone la implementación de una biblioteca de algoritmos para la generación automática del horario docente correspondiente a la Facultad 15 de la Universidad de las Ciencias Informáticas. Surge a partir de la necesidad de automatizar el proceso de planificación docente que en ese momento se realizaba de forma manual y no se ajustaba al dinamismo de la universidad.

Se realiza la investigación y construcción heurística para la obtención de diferentes horarios factibles, así como la optimización de los resultados iniciales a través de la metaheurística implementada. Genera finalmente el horario docente de acuerdo a las restricciones y condiciones de la Facultad 15 y tiene en cuenta la variabilidad de los recursos humanos y de infraestructura, permite reducir el tiempo y los recursos invertidos en la realización del proceso (González, 2010).

Durante el año 2012 se desarrolló la investigación de la tesis titulada **Sistema de Gestión para la planificación docente en la Facultad Regional “Mártires de Artemisa”**, entre los objetivos generales de la investigación está desarrollar un sistema de gestión para la planificación docente del Departamento Docente General de la Facultad Regional “Mártires de Artemisa”, el cual contribuya a mejorar la integridad

de la información que se maneja en el mismo. El Sistema que se desarrolla en la anteriormente mencionada investigación posee nuevas funcionalidades y podrá implantarse en cualquier institución que lo necesite. Además, con su desarrollo se lograría un mejor aprovechamiento de las tecnologías existentes en la Facultad Regional “Mártires de Artemisa”, y una mejor manipulación de la documentación que se genera en dicho Departamento (Guerra, 2012).

Se analizó además como parte de la investigación del objeto de estudio la tesis titulada: **Análisis y Diseño de un sistema para la planificación automatizada del Horario Docente de la Facultad 4**, que cómo se puede apreciar en el título de este trabajo de diploma, los autores solo se limitaron al análisis y diseño del sistema que se deseaba obtener, faltando el desarrollo e implementación del mismo.

Se tuvo en cuenta también, el Módulo de Gestión de Horarios de **Akademos**, el cual no es más que un sistema automatizado para la gestión académica desarrollada por un equipo de trabajo de la Dirección de Informatización de la UCI, que permite la gestión automatizada de los elementos que intervienen en la labor académica, además, da la opción de mitigar los riesgos que imponen el dinamismo del proceso de gestión académica enfrentando de forma natural los cambios requeridos y adaptándose a las nuevas condiciones (Costa, 2007). El presente sistema era utilizado para la gestión del Horario Docente de la Universidad, pero teniendo en cuenta que se realizaba la generación del mismo de manera manual.

Además, se llevó a cabo el análisis y diseño del Sistema de Planificación de Horarios utilizando RUP como metodología de desarrollo (Ramos y Iglesias, 2008). Entre las principales funcionalidades que presenta este sistema, según (Ramos y Iglesias, 2008) están:

- ✚ Gestionar Horario: Para confeccionar el horario se registran los siguientes datos: profesores, asignaturas, grupos, semanas, locales, balance de carga.
- ✚ Gestionar Balance de Carga: De cada balance de carga se registran los siguientes datos: asignatura, semana, frecuencias semanales, tipo de clase (clase práctica, laboratorio, conferencia), orden.

En la Tabla 1.2 podemos observar que los sistemas y módulos estudiados a nivel nacional no cumplen con todos los requisitos necesarios para dar solución al problema planteado en la presente investigación, uno de ellos solo se limitó a analizar y diseñar el sistema para la planificación automatizada del Horario Docente, y los demás no están implementados completamente. A pesar de que uno de ellos es de Tipo de Software

Libre, y además genera de manera automática el Horario Docente no se puede utilizar para dar solución al problema planteado pues no trae consigo en su funcionamiento y opciones la posibilidad de reajustar el horario docente creado a raíz de alguna afectación extra docente de algunos de los profesores encargados de impartir algún turno de clase específico.

Tabla 1.2 Resumen comparativo de los Sistemas y Módulos de Gestión de Horarios Nacionales.

Parámetros	Biblioteca de Algoritmos	Facultad Regional Mártires de Artemisa	Facultad 4	Akademos
Tipo de Software Libre	-	Sí	-	Sí
Licencia de Pago	No	No	-	No
Generación automática del horario	Sí	Sí	-	No
Tecnología de desarrollo	IA	Java	No	-
Afectaciones docentes	-	No	No	-
Flexibilidad	-	-	No	-

1.3 Metodología, herramientas y tecnologías para dar solución al problema

En el epígrafe que se presenta a continuación se puede apreciar la metodología, herramientas y tecnologías a utilizar durante la implementación, desarrollo y prueba de la propuesta de solución, así como una breve descripción de cada una de ellas. La elección de cada una de las herramientas y tecnologías antes mencionadas fueron en consenso por parte del equipo de desarrollo y teniendo en cuenta el proceso de desarrollo de software de la Universidad de las Ciencias Informáticas.

Metodología a usar durante el desarrollo de la Investigación

Una metodología de desarrollo de software se puede definir como un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda en la construcción de un software. Actualmente, existen diversas metodologías recogidas en dos grandes grupos, las denominadas metodologías tradicionales o formales y las ágiles o ligeras. Las metodologías ágiles se caracterizan por ser más orientadas al desarrollo de software, con bajos niveles de formalización en la documentación requerida y por ser, a diferencia de las tradicionales, más adaptables a los cambios, requerir de pequeños grupos de trabajo y por ser apropiadas para entornos volátiles (Romero y Castrillón, 2019).

Para la presente investigación se ha decidido utilizar la metodología ágil por todo lo antes expuesto, específicamente la AUP realizando una variación de la misma de manera tal que se adapte a las especificaciones de la investigación y a la actividad productiva de la UCI, ya que esta es la que se utiliza en la mayoría de procesos de desarrollo de software de la Universidad, y se aplica a los proyectos que modelan exclusivamente los conceptos fundamentales del negocio por lo que es recomendado para proyectos donde el objetivo primario es la gestión y presentación de información como es el caso de la presente investigación.

Entre los principales artefactos de la metodología a utilizar, la presente investigación se centrará en los correspondientes al escenario 2. Para dicho escenario se tiene presente el Modelo Conceptual, y los Casos de Uso del Sistema como artefactos principales.

Una vez analizados las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, lo que conlleva a que se unifican las restantes 3 fases de AUP en una sola, a la que llamaremos Ejecución y se agrega la fase de Cierre (Rodríguez, 2015).

El proceso de desarrollo de software de la UCI actualmente se encuentra certificado en el nivel 2 de ²CMMI, algo que sin lugar a dudas brinda un salto de calidad y confianza a los programas y softwares desarrollados en la Universidad. Según (Rodríguez y López, 2019), el principal propósito de CMMI es mejorar los procesos que se desarrollan ya sea en una empresa, compañía, o alguna institución, así como brindar una guía para el proceso de mejora y habilidad para manejar el desarrollo, adquisición y mantenimiento de los productos o servicios dentro de la institución.

Lenguaje Unificado de Modelado UML 2.5.1

El Lenguaje Unificado de Modelado (UML) es, tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso, está compuesto por una notación muy específica y por las reglas semánticas relacionadas para la construcción de sistemas de software. El UML en sí mismo no prescribe ni aconseja cómo usar esta notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos, sin embargo, soporta un conjunto rico en elementos de notación gráficos. Describe la notación

² Es un modelo de mejora del rendimiento de clase mundial para organizaciones competitivas que deseen lograr operaciones de alto rendimiento. CMMI consiste en recolectar las mejores prácticas diseñadas para promover los comportamientos que conducen a un mejor desempeño en cualquier organización.

para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos. Además, soporta la idea de extensiones personalizadas a través de elementos estereotipados, permite beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo (Sparks, 2022).

Según (Chaparro y Vicente, 2020), el Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medias. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notaciones, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código, así como generadores de informes. La especificación de UML no define un proceso estándar, pero está elaborado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.

Visual Paradigm 8.0

Visual Paradigm (anteriormente VP-UML) es una herramienta UML. La herramienta está diseñada para una amplia gama de usuarios, incluidos ingenieros de software, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquiera que esté interesado en construir de manera confiable sistemas de software a gran escala utilizando un enfoque orientado a objetos. Además, Visual Paradigm admite los últimos estándares de notación UML (Visual Paradigm, 2022).

Entre las principales características se pueden resaltar que es una potente herramienta de modelado visual que ayuda a crear y administrar los diagramas y elementos del modelo, además como ventajas principales se pueden encontrar (Visual Paradigm, 2022):

- Es un gran editor de diagramas de arrastrar y soltar.
- Admite UML, BPMN, ArchiMate, DFD, ERD, SoaML, SysML, CMMN.

- Es una herramienta de modelado eficaz, utiliza la reutilización de elementos, la transformación de diagramas y elementos, la validación de sintaxis, las propiedades personalizadas, etc.
- Presenta muchas opciones de formato.

PHP 8.0.3

PHP es un popular lenguaje de secuencias de comandos de propósito general que es especialmente adecuado para el desarrollo web. Rápido, flexible y pragmático, PHP potencia todo, desde su blog hasta los sitios web más populares del mundo. PHP se centra principalmente en las secuencias de comandos del lado del servidor, por lo que puede hacer cualquier cosa que pueda hacer cualquier otro programa CGI, como recopilar datos de formulario, generar contenido de página dinámico o enviar y recibir cookies. Pero PHP puede hacer mucho más (PHP, 2022).

Hay tres áreas principales donde se utilizan scripts PHP:

Scripting del lado del servidor: Este es el campo de destino más tradicional y principal para PHP. Necesita tres cosas para que esto funcione: el analizador PHP (CGI o módulo de servidor), un servidor web y un navegador web. Debe ejecutar el servidor web, con una instalación PHP conectada. Puede acceder a la salida del programa PHP con un navegador web, viendo la página PHP a través del servidor. Todos estos pueden ejecutarse en su máquina doméstica si solo está experimentando con la programación PHP. Consulte la sección de instrucciones de instalación para obtener más información.

Secuencias de comandos de línea de comandos: Puede hacer un script PHP para ejecutarlo sin ningún servidor o navegador. Solamente necesita el analizador PHP para usarlo de esta manera. Este tipo de uso es ideal para secuencias de comandos que se ejecutan regularmente con cron o el programador de tareas (en Windows). Estos scripts también se pueden usar para tareas simples de procesamiento de texto. Consulte la sección sobre el uso de la línea de comandos de PHP para obtener más información.

Escritura de aplicaciones de escritorio: PHP probablemente no sea el mejor lenguaje para crear una aplicación de escritorio con una interfaz gráfica de usuario, pero si se conoce PHP muy bien y le gustaría utilizar algunas características avanzadas de PHP en sus aplicaciones del lado del cliente, también puede usar PHP-GTK para escribir este tipo de programas.

PHP se puede usar en todos los principales sistemas operativos, incluido Linux, muchas variantes de Unix, Microsoft Windows, macOS, RISC OS y probablemente otros. PHP también tiene soporte para la mayoría de los servidores web actuales.

JavaScript

JavaScript nació en el año 1995, prácticamente de la mano de la WWW, o Internet Comercial. Fue diseñado en un principio por Netscape Communications. Integrado originalmente como una especie de plugin en los navegadores de la primera Web, fue muy criticado por su lentitud para procesar código, dado que esta acción generaba un retardo notable en la carga completa de una página. A los pocos años de vida fue destronado por Flash Player, pero tan solo una década después, volvió completamente recargado, y recuperó su lugar y labor dentro del desarrollo Web (Luna, 2019).

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, entre otros. Es la tercera capa del pastel de las tecnologías web estándar, dos de las cuales son HTML y CSS (JavaScript, 2022).

Entre los principales usos de JavaScript se encuentran (jvatpoint, 2022):

- Aplicaciones Web
- Desarrollo Web
- Aplicaciones Móviles
- Juegos
- Presentaciones
- Aplicaciones de Servidor
- Servidores Web

HTML 5

El nacimiento de HTML5 constituye un punto de inflexión en el modo que se desarrollan los estándares web. Su origen radica en un pequeño grupo de profesionales disconformes con el rumbo que había decidido

tomar la organización que vela por el desarrollo de los estándares. Un grupo de desarrolladores de la Fundación Mozilla y de Opera Software, a los que se les unen más tarde profesionales de Apple, fundan en 2004 el Grupo de Trabajo de Tecnología de Aplicaciones de Hipertexto Web (Web Hypertext Application Technology Working Group) al margen del W3C (Gutiérrez, 2016).

HTML5 es un estándar que sirve como referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado HTML) para la definición de contenido de una página web, como texto, imágenes, vídeos, juegos, entre otros. HTML 5 propone estándares para cada aspecto de la web y además un propósito claro para cada una de las tecnologías involucradas. Cada parte del documento está diferenciada, declarada y determinada por etiquetas específicas (HTML5, 2019).

CSS 3

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas (Álvarez, 2017).

Con CSS 3 hemos obtenido una actualización importante en el estándar de definición de estilos para documentos HTML, que incluye características potentes, tanto para aplicar aspecto avanzado en elementos de una página como para ayudarnos a realizar una maquetación más precisa. El objetivo final es que los desarrolladores sean capaces de aplicar estilos a los documentos HTML de manera versátil, sin necesidad de hacks y facilitando la separación entre contenido y presentación. CSS 3 ya no es una novedad, sino una versión del lenguaje totalmente adoptada por los navegadores, que todo desarrollador debe utilizar en su día a día (Manual de CSS 3, 2022).

A diferencia de CSS 2, que fue una gran especificación que definía varias funcionalidades, CSS 3 está dividida en varios documentos separados, llamados "módulos". Cada módulo añade nuevas funcionalidades a las definidas en CSS 2, de manera que se preservan las anteriores para mantener la compatibilidad. Debido a la modularización del CSS 3, diferentes módulos pueden encontrarse en diferentes estados de su desarrollo, existiendo actualmente alrededor de cincuenta módulos publicados. Mientras CSS 3 se convierte en un estándar, los distintos navegadores pueden incluir o no las mejoras propuestas, o incluso añadir las suyas propias (Introducción | CSS 3, 2022).

Framework Laravel 9, Bootstrap 5

Laravel es un marco de aplicación web con una sintaxis expresiva y elegante. Un marco web proporciona una estructura y un punto de partida para crear su aplicación, lo que le permite concentrarse en crear algo increíble mientras nos preocupamos por los detalles. Laravel se esfuerza por brindar una experiencia de desarrollador increíble, al mismo tiempo que brinda características poderosas, como una inyección de dependencia completa, una capa de abstracción de base de datos expresiva, colas y trabajos programados, pruebas unitarias y de integración, y más. Laravel combina los mejores paquetes en el ecosistema de PHP para ofrecer el marco disponible más robusto y fácil de usar para desarrolladores. Además, miles de talentosos desarrolladores de todo el mundo han contribuido al marco (Laravel, 2022).

Bootstrap es un kit de herramientas frontal potente, ampliable y repleto de funciones. Se puede utilizar para producir y personalizar con Sass, utiliza el sistema de grillas y los componentes prediseñados y permite que los proyectos cobren vida con potentes complementos de JavaScript (Bootstrap, 2022).

Bootstrap es un producto de código abierto de Mark Otto y Jacob Thornton quienes, cuando se lanzó inicialmente, eran empleados de Twitter. Desde que se lanzó Bootstrap en agosto de 2011, ha ganado popularidad. Ha evolucionado de ser un proyecto completamente basado en CSS para incluir una gran cantidad de complementos e íconos de JavaScript que van de la mano con formularios y botones. Todo esto permite que el desarrollo web frontal sea catapultado hacia adelante, construyendo sobre una base estable de diseño y desarrollo con visión de futuro (Spurlock, 2013).

Son varias las ventajas que nos ofrece Bootstrap a la hora de utilizarlo, como por ejemplo (GAIKWAD, 2019):

- Primer enfoque móvil, el marco consiste en los primeros estilos móviles en toda la biblioteca en lugar de ellos en archivos separados.
- Compatibilidad con navegador: es compatible con todos los navegadores.
- Fácil de empezar, con solo el conocimiento de HTML y CSS con los que cualquiera puede empezar.
- Diseño responsive – CSS responsive de Bootstrap se ajusta a Desktops, Tablets y Móviles.
- Es que es de código abierto.

Visual Studio Code Versión 1.67

Visual Studio Code es un editor de código fuente ligero, pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y

Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (Visual Studio Code, 2022).

La depuración suele ser la característica que más extrañan los desarrolladores en una experiencia de codificación más eficiente, así que lo hicimos posible. Visual Studio Code incluye un depurador interactivo, por lo que puede recorrer el código fuente, inspeccionar variables, ver pilas de llamadas y ejecutar comandos en la consola. También se integra con herramientas de compilación y secuencias de comandos para realizar tareas comunes y acelerar los flujos de trabajo diarios, es compatible con Git, por lo que puede trabajar con control de código fuente sin salir del editor, incluida la visualización de diferencias de cambios pendientes (Visual Studio Code, 2022).

XAMPP 3.3.0

XAMPP es una abreviatura en la que la X significa "Cross-Platform" (plataforma cruzada), la A significa "Apache", la M significa "MYSQL" y la P significa "PHP" y "Perl", respectivamente, es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar. XAMPP proporciona un entorno de desarrollo local ideal, con la intención de lograr hacer que el alojamiento de aplicaciones PHP creadas con XAMPP sea lo más fácil posible. Entre las principales ventajas de las características del servidor XAMPP se puede destacar que la mayor ventaja que tiene sobre cualquier otro servidor web es que es fácil de configurar y usar, además es un software multiplataforma disponible para todo tipo de sistemas operativos como Linux y Windows. También posee muchos otros módulos esenciales como phpMyAdmin, OpenSSL, MediaWiki, WordPress, Joomla y algunos más, por otra parte, podemos encontrar como desventaja principal que la configuración es muy difícil de realizar en comparación con otros programas de similar funcionamiento, como es caso de Wamp (XAMPP, 2022).

Apache JMeter 5.4.1

La aplicación Apache JMeter es un software de código abierto, una aplicación 100% pura de Java diseñada para cargar el comportamiento funcional y medir el rendimiento. Se diseñó originalmente para probar aplicaciones web, pero desde entonces se ha ampliado a otras funciones de prueba. Apache JMeter puede utilizarse para probar el rendimiento tanto de los recursos estáticos como de los

dinámicos, de las aplicaciones web dinámicas. Se puede utilizar para simular una carga pesada en un servidor, grupo de servidores, red u objeto para probar su resistencia o para analizar el rendimiento general bajo diferentes tipos de carga (Apache JMeter, 2022).

Acunetix Web Vulnerability Scanner 14.8

Acunetix no es sólo un escáner de vulnerabilidad web, es una solución completa de pruebas de seguridad de aplicaciones web que puede utilizarse tanto de forma autónoma como como parte de entornos complejos. Ofrece evaluación y gestión de vulnerabilidades incorporadas, así como muchas opciones de integración con herramientas de desarrollo de software líderes en el mercado. Al hacer de Acunetix una de sus medidas de seguridad, puede aumentar significativamente su postura de ciberseguridad y eliminar muchos riesgos de seguridad con un bajo coste de recursos (Vulnerability Scanner, 2022).

CONCLUSIONES PARCIALES DEL CAPÍTULO

En el presente capítulo se han abordado los principales elementos teóricos que permiten dar base y sustento a la gestión de las actividades docentes a partir de la generación de horarios desde el Sistema de Trabajo Educativo. El estudio del arte realizado a lo largo del capítulo contribuyó a la recopilación de información y tendencias acerca de los sistemas y módulos de planificación y confección de horarios docentes analizados y observados, lo que permitió concluir que a pesar de que existen sistemas capaces de satisfacer algunas de las necesidades que imponen el problema de investigación, no se pueden utilizar debido a que principalmente no cumplen con los requisitos necesarios. El análisis de las herramientas y las tecnologías empleadas durante el capítulo han permitido definir y abordar un poco más sobre el desarrollo de la propuesta de solución del problema de investigación, apoyándose siempre en el escenario 2 de la metodología ágil AUP-UCI y sus artefactos.

CAPÍTULO 2: ANÁLISIS Y DISEÑO DEL MÓDULO DE GENERACIÓN DEL HORARIO DOCENTE PARA EL SISTEMA DE TRABAJO EDUCATIVO EN LA FACULTAD 1

En el presente capítulo se realiza una descripción de la propuesta de solución mediante los requisitos funcionales y no funcionales principalmente, así como los elementos particulares del diseño de la misma. Además, se pueden observar aspectos esenciales de la arquitectura a utilizar durante el desarrollo del módulo y los patrones de diseño empleados. Por otra parte, se describen los artefactos para el análisis y diseño que se generan durante el flujo principal de trabajo.

2.1 Descripción actual del proceso de planificación de Horarios Docentes en la Facultad 1.

En la Universidad de las Ciencias Informáticas (UCI), se realiza la planificación y confección del horario docente de la Facultad 1 para cada semana y semestre del curso, al realizarse el mismo por una persona encargada de hacerlo manualmente pueden sufrir cambios en la distribución de las clases de los años de la Facultad, y a su vez se torna a ser un poco complejo su creación.

Para la correcta elaboración del Horario Docente, los profesores principales de cada año o la Vicedecana, necesitan tener disponibles ciertos documentos y requisitos tales como: Asignaturas, Balance de carga en dependencia del año docente, Carga Docente, entre otros. Es importante también tener en cuenta las disponibilidades de locales (Salones de Conferencia, Laboratorios, Aulas), así como considerar las posibles afectaciones de algunos profesores por ciertos problemas ajenos a su voluntad o que presenten alguna condición específica que le impida asistir el día que le corresponde impartir algún turno de clase en específico. Una vez que el profesor(a) principal del año o la Vicedecana, confecciona el horario docente, teniendo en cuenta todas las restricciones posibles que existen, la planificadora debe revisar el horario generado, de manera que compruebe que no exista conflictos con los locales en los que se planificaron las clases, así como que se cumplan con las afectaciones docentes que puedan presentar los profesores. Luego de estar listo el horario docente se debe compartir tanto con estudiantes, como con los profesores de la facultad.

En la Figura 1. Modelo Conceptual, se puede observar el modelo conceptual del negocio referente a la confección del horario docente de la Facultad 1.

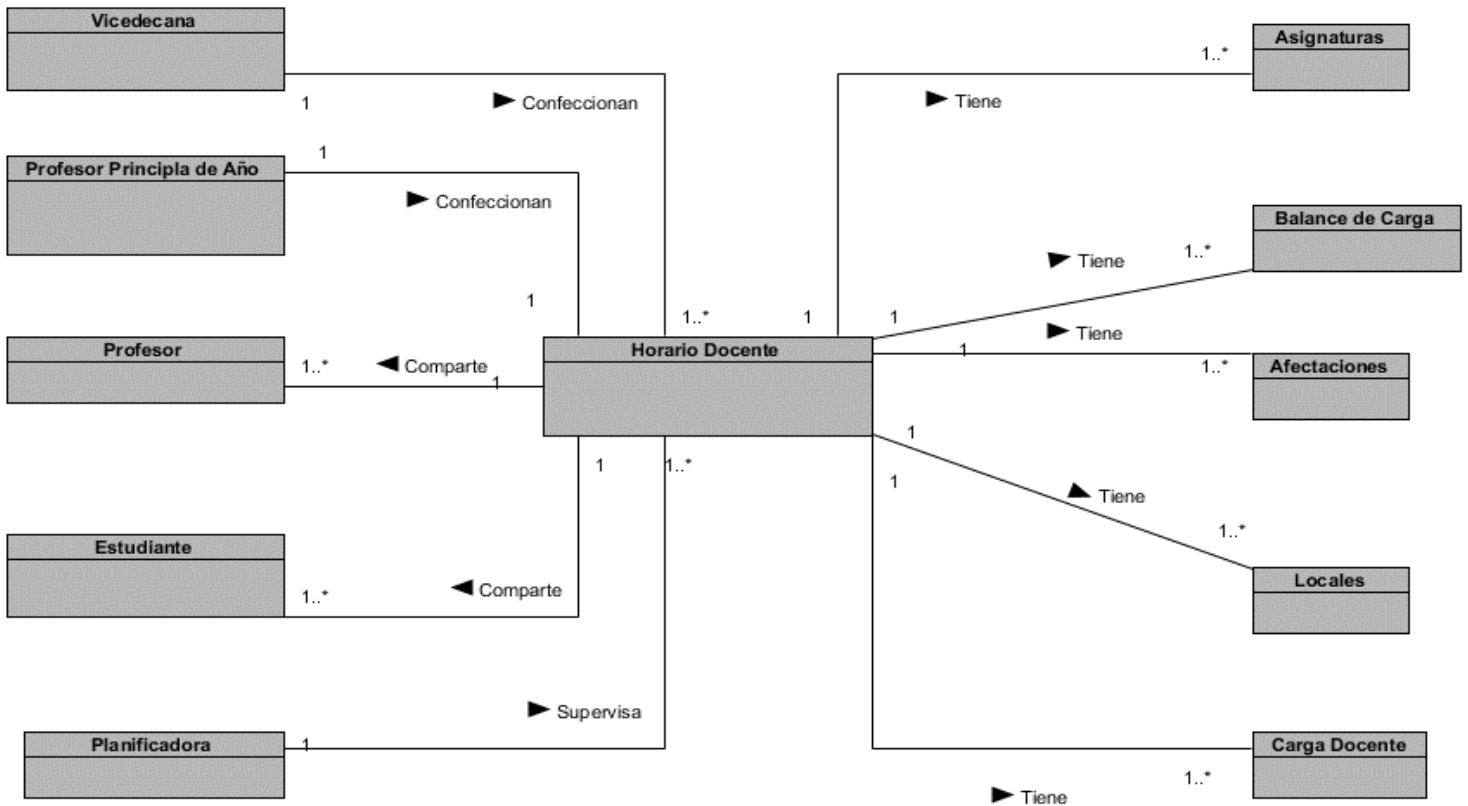


Figura 1. Modelo Conceptual

A continuación, se detalla el significado de los conceptos representados en el modelo conceptual:

Conceptos:

- Vicedecana: Profesor(a) de la Facultad con permisos para generar el horario docente de todos los años de la Facultad.
- Profesor Principal de Año: Profesor(a) Principal de cada año con permisos para generar el horario docente de su año.
- Profesor: Profesores de la Facultad que pueden visualizar el horario docente.
- Estudiante: Estudiantes de la Facultad que pueden visualizar el horario docente.
- Planificadora: Profesor(a) de la Facultad que supervisa el horario docente de todos los años una vez confeccionados.
- Horario Docente: Horario docente de cada año de la facultad.

- Asignaturas: Grupo de asignaturas a impartir de cada año docente.
- Balance de carga: Balance de carga de las asignaturas a impartir por año docente
- Afectaciones: Posibles afectaciones docentes, que impidan a algún profesor específico impartir un turno de clases.
- Locales: Todos los locales de la Facultad disponibles para impartir turnos de clases.
- Carga Docente: Relación entre los profesores, las asignaturas y los grupos por año docente.

2.2 Descripción de la propuesta de solución para la generación de horarios docentes.

Es necesario analizar y definir las principales características que abarcan de manera general el módulo de generación del horario docente de la Facultad 1, así como los conceptos fundamentales asociados al proceso de planificación y confección del Horario Docente de la Facultad.

El módulo para la planificación y confección del horario docente para el Sistema de Trabajo Educativo de la Facultad 1, presenta como característica principal la posibilidad de la confección de manera automática del Horario Docente. Para lograr esto, el módulo estará integrado a un Sistema de Trabajo Educativo (STE) que se está desarrollando actualmente en la Facultad 1, con el objetivo de que dentro de dicho sistema se cree el horario docente por los profesores principales de cada año o por la Vicedecana de la Facultad. Para lograr la correcta confección del horario docente, el módulo permitirá la gestión de distintos datos necesarios para el proceso, tales como: Asignaturas, Afectaciones Docentes, Balance de Carga, Locales, Pruebas Parciales y la Carga Docente. Una vez confeccionado el horario se puede compartir con los estudiantes y profesores de la Facultad.

2.3 Análisis de la propuesta de solución para la confección del horario docente.

Los requisitos funcionales y no funcionales tienen como objetivo principal lograr que un sistema de software funcione con eficacia. Cada requisito lógicamente tiene características únicas que contribuyen a aumentar la usabilidad y la seguridad, ya que permiten que el software funcione más rápido y utilice protocolos de seguridad sólidos. Por otra parte, utilizar algunas de las tantas herramientas y vías de modelado de software existentes permite mejorar la productividad del equipo de desarrollo, lo que a su vez facilita la comprensión y la documentación del sistema a desarrollar para un futuro (Carrizo, 2018).

2.3.1 Técnicas de Captura de Requisitos

Obtener y definir requisitos en etapas tempranas de un proyecto de software es una tarea compleja, sin embargo, saber en qué áreas de la ingeniería de requisitos existen mayor cantidad de estudios y qué tipo de ellos se han identificado, es crucial para el desarrollo de nuevas investigaciones y búsqueda de soluciones según (Carrizo, 2018). Con el objetivo de conocer el alcance de la propuesta de solución se llevaron a cabo varias técnicas de obtención de requisitos:

- **Entrevista:** Ayudó a comprender con mayor objetividad el problema a informatizar, y él porque es necesario el desarrollo de dicha solución. Para esto, se realizaron entrevistas a varios profesores principales de año, y otros cargos de la facultad encargados de utilizar toda la información referente a la planificación y confección del Horario Docente de la Facultad 1 (Ver Anexo 2).
- **Análisis de los Sistema Homólogos Nacionales:** Como fuente importante para la obtención de requisitos principales del módulo a desarrollar, se encuentra el análisis a fondo de los sistemas homólogos estudiados y antes descritos en el epígrafe 1.2.2.

2.3.2 Requisitos Funcionales

Los requisitos funcionales son generalmente relacionados con los servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo debería comportarse en situaciones específicas. En algunos casos variados, los requerimientos funcionales también explican lo que el sistema no debe hacer (Herrera, 2017).

Luego del encuentro realizado con el cliente, se obtuvo un total de 32 requisitos funcionales, para los cuales se decidió asignar una prioridad específica teniendo en cuenta la importancia decidida por el cliente a partir de las principales necesidades. En la siguiente Tabla 2, se especifican los mismos:

Tabla 2. Requisitos Funcionales

Requisito	Prioridad	Descripción	Requisito	Prioridad	Descripción
RF1: Insertar Local	Alta	Permite al usuario con rol Profesor Principal de Año o	RF2: Eliminar Local	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana

		Vicedecana insertar un local.			eliminar un local.
RF3: Editar Local	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana editar un local.	RF4: Buscar Local	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana buscar un local.
RF5: Listar Local	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana observar la lista de los locales insertados.	RF6: Insertar Asignatura	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana insertar una asignatura.
RF7: Eliminar Asignatura	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana eliminar una asignatura.	RF8: Editar Asignatura	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana editar una asignatura.
RF9: Buscar Asignatura	Media	Permite al usuario con rol Profesor Principal de	RF10: Listar Asignatura	Media	Permite al usuario con rol Profesor Principal de Año

		Año o Vicedecana buscar una asignatura.			o Vicedecana observar la lista de asignaturas insertadas.
RF11: Insertar Afectación	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana insertar una afectación.	RF12: Editar Afectación	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana editar una afectación.
RF13: Eliminar Afectación	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana eliminar una afectación.	RF14: Buscar Afectación	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana buscar una afectación.
RF15: Listar Afectación	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana observar la lista de afectaciones insertadas.	RF16: Visualizar Horario Docente	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana o Estudiante o Profesor, visualizar el Horario Docente.

RF17: Insertar Balance de carga	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana insertar datos en el Balance de Carga.	RF18: Editar Balance de carga	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana editar datos en el Balance de Carga.
RF19: Eliminar Balance de carga	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana eliminar datos en el Balance de Carga.	RF20: Listar Balance de Carga	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana observar los datos insertados en el Balance de Carga.
RF21: Generar Horario	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana generar un horario docente.	RF22: Buscar Balance de Carga	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana buscar datos en el Balance de Carga.
RF 23: Exportar Balance de Carga como Excel	Alta	Permite al usuario con rol Profesor Principal de Año o	RF24: Listar Carga Docente	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana

		Vicedecana exportar el Balance de Carga en formato Excel.			observar los datos insertados en la Carga Docente.
RF25: Insertar Carga Docente	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana insertar la Carga Docente.	RF26: Editar Carga Docente	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana editar la Carga Docente.
RF27: Eliminar Carga Docente	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana eliminar la Carga Docente.	RF28: Buscar Carga Docente	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana buscar datos de la Carga Docente.
RF29: Insertar Prueba Parcial	Alta	Permite al usuario con rol Profesor Principal de Año o Vicedecana insertar una prueba parcial.	RF30: Buscar Prueba Parcial	Media	Permite al usuario con rol Profesor Principal de Año o Vicedecana buscar una prueba parcial ya insertada.
RF31: Eliminar Prueba Parcial	Alta	Permite al usuario con rol Profesor Principal de	RF32: Listar Pruebas Parciales	Media	Permite al usuario con rol Profesor Principal de Año

		Año o Vicedecana eliminar una prueba parcial.			o Vicedecana visualizar un listado con las pruebas parciales insertadas.
--	--	--	--	--	---

2.3.3 Requisitos No Funcionales

La mayoría de los softwares presentan un cierto número de atributos asociados que reflejan la calidad de dicho software. Estos atributos antes mencionados no están directamente asociados con lo que el software como tal hace, más bien, reflejan su comportamiento durante su ejecución y en la estructura y organización del programa fuente y en la documentación asociada. Los requisitos no funcionales son por lo general restricciones de los servicios o funciones ofrecidos por el sistema, que incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales como también lo llaman otros autores, a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (SOMMERVILLE, 2005).

Para la presente investigación el autor considera los requisitos no funcionales (RnF) como requerimientos de calidad, los cuales se asumen directamente desde el Sistema de Trabajo Educativo de la Facultad 1, debido a que el módulo de generación de horarios docentes que se desea generar estará integrado en el propio Sistema. Se dividen en varias especificaciones como: usabilidad, confiabilidad, portabilidad eficiencia, y mantenibilidad.

- **Usabilidad:**

RnF 1: El módulo debe presentar una interfaz agradable y de buen gusto para el usuario.

RnF 2: El módulo podrá ser visualizado en dispositivos desde las resoluciones 768x1024 y 1366x768.

- **Confiabilidad:**

RnF 3: El módulo debe mostrar solo la información necesaria, de manera que el usuario logre estar orientado todo el tiempo.

- **Portabilidad:**

RnF 4: La propuesta de solución se debe ejecutar en los navegadores existentes en la Universidad actualmente (Chrome, Mozilla Firefox), así como adaptar su interfaz a cualquier dispositivo que acceda al módulo.

- **Seguridad:**

RnF 5: Se garantizará la integridad de la información mediante mecanismos de control de acceso, utilizando usuario, contraseña y niveles de accesos para cada usuario de acuerdo a su rol, asegurando que cada uno pueda tener disponible solamente las opciones que se encuentran en correspondencia con su actividad.

RnF 6: El módulo debe gestionar y requerir información de usuarios para su uso.

RnF 7: La información manejada por el módulo está protegida con el objetivo de evitar accesos no autorizados.

RnF 8: Ante los posibles errores que puedan ocurrir en el módulo, no se debe mostrar detalles específicos de información que pueda comprometer su seguridad.

RnF 9: El módulo deberá ser capaz de cerrar la sesión del usuario una vez pasado un determinado tiempo de inactividad.

- **Apariencia o Interfaz externa:**

RnF 10: Empleo de colores claros e interfaces sencillas para el usuario.

- **Eficiencia:**

RnF 11: El tiempo de demora de una petición al servidor debe ser menor de 8 a 10 segundos aproximadamente.

- **Mantenibilidad:**

RnF 12: Se debe permitir realizar modificaciones posteriores para mejorar el módulo o en caso de que existan nuevas exigencias por parte del cliente.

RnF 13: El módulo debe estar bien documentado de manera tal que el tiempo de mantenimiento sea lo más pequeño posible.

- **Hardware:**

RnF 14: El servidor de base de datos debe poseer una capacidad mínima de 20 GB.

RnF 15: El servidor de aplicaciones web debe poseer una capacidad mínima de 80 GB.

RnF 16: Los servidores web y de base de datos deben poseer como mínimo 2 GB de memoria RAM.

RnF 17: Se requiere una tarjeta de red con velocidad de 100 Mbps.

RnF 18: Se requiere 2 GB de memoria RAM como mínimo.

RnF 19: Se requiere un procesador de 2 GHZ como mínimo.

- **Software:**

RnF 20: Un navegador Chrome o Mozilla Firefox.

RnF 21: Sistema operativo Linux o Windows 10.

RnF 22: Servidor Web Apache con PHP 8.0.3

RnF 23: MySQL 8.0 como gestor de bases de datos.

2.3.4 Diagrama de Caso de Uso del Sistema

El diagrama de casos de uso, es una forma de diagrama de comportamiento en lenguaje de modelado unificado, con la que se representan procesos empresariales, así como sistemas y procesos de programación orientada a objetos, en este diagrama, todos los objetos involucrados se estructuran y se relacionan entre sí (Desarrollo Web, 2022).

La Figura 2. Diagrama de Caso de uso del Sistema, muestra el diagrama de caso de uso del sistema de la presente investigación.

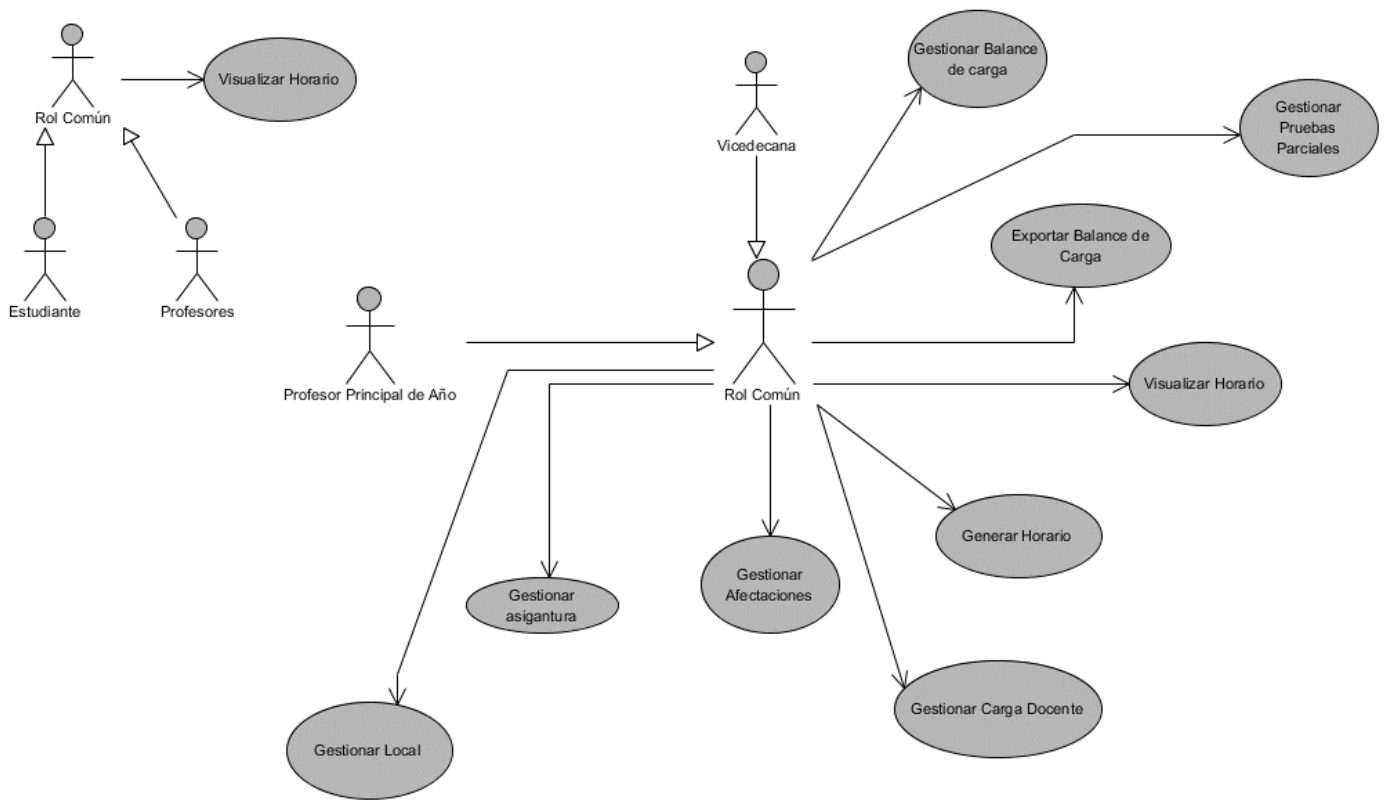


Figura 2 Diagrama de Casos de Usos del Sistema.

2.3.5 Descripción de Casos de Uso del Sistema

CU 1. <Gestionar Asignatura>

Objetivo	Permite insertar, modificar, eliminar, listar y buscar los datos de una asignatura.
Actores	Profesor Principal de Año y Vicedecana.
Resumen	El caso de uso inicia cuando el Profesor Principal o la Vicedecana Inserta, modifica y elimina los datos de una asignatura.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Profesor Principal de Año o Vicedecana ya autenticado.
Postcondiciones	Se registró, modificó o eliminó una asignatura.
Flujo de eventos	
Flujo básico < Gestionar Asignatura >	

Actor		Sistema
1.	Selecciona la opción "Asignatura".	
2.		Muestra en pantalla una lista con las opciones de Insertar Asignatura, y Editar o Eliminar si ya existen datos insertados en la lista, además del listado de las asignaturas ya insertadas.
3.	Desea Insertar, Editar o Eliminar una asignatura.	
4.		<p>El sistema despliega una pantalla con las opciones para la gestión de asignaturas.</p> <p>a) Para insertar asignaturas ir a la sección "Insertar asignatura"</p> <p>b) Para eliminar asignaturas ir a la sección "Eliminar Asignatura"</p> <p>c) Para modificar asignaturas ir a la sección "Editar Asignatura"</p>

Flujos alternos

2a. Listado de asignaturas vacías.

Actor		Sistema
1.		Muestra en pantalla un Listado Vacío.

Sección 1: "Insertar asignatura"

Flujo básico < Insertar asignatura >

Actor		Sistema
1.	Presiona la opción "Insertar asignatura".	
2.		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> -Sesión -Nombre, -Año Docente

		-Semestre -Estado Y el botón "Guardar". Y el botón "Cancelar".
3.	Introduce los datos de la asignatura en los campos y presiona la opción "Guardar" o "Cancelar".	
4.		Verifica que los campos estén llenos.
5.		Verifica que la asignatura no esté registrada.
6.		Guarda los datos de las asignaturas y muestra un mensaje "Asignatura registrada". Finalizando así el Caso de Uso.
Flujos alternos		
4a. Campos vacíos.		
Actor		Sistema
1.		Muestra un mensaje "Los Campos son Requeridos".
Flujos alternos		
5a. Asignatura registrada.		
Actor		Sistema
1.		Muestra un mensaje "Asignatura registrada".
Sección 2: "Editar asignatura"		
Flujo básico < Editar asignatura >		
Actor		Sistema
1.	Presiona la opción "Editar" en la lista de asignaturas mostradas.	
2.		Muestra un formulario con los siguientes campos a modificar: -Sesión -Nombre,

		-Año Docente -Semestre -Estado Y el botón "Editar". Y el botón "Cancelar".
3.	Actualiza los datos de las asignaturas en los campos y presiona la opción "Guardar" o "Cancelar".	
4.		Verifica que los campos estén llenos.
5.		Actualiza los datos de las asignaturas y muestra un mensaje "Asignatura editada". Finalizando así el Caso de Uso.
Flujos alternos		
4a. Campos vacíos.		
Actor		Sistema
1.		Muestra un mensaje "Campos vacíos".
Sección 3: "Eliminar asignatura"		
Flujo básico <Eliminar asignatura >		
Actor		Sistema
1.	Presiona el botón 'Eliminar' en la Asignatura que se desea eliminar.	
2.		Muestra en pantalla un mensaje "Esta seguro que desea eliminar". -y el botón "Aceptar" -Y el botón "Cancelar"
3.	Presiona el botón "Aceptar "	
4.		Elimina los datos de las asignaturas y muestra un mensaje "Asignatura eliminada". Finalizando así el Caso de Uso.
Flujos alternos		

2a. Marca botón 'Cancelar'		
Actor		Sistema
1.		Retornar al paso 2 en la sección "Asignatura"
Sección 4: "Buscar Asignatura"		
Flujo básico <Buscar Asignatura >		
Actor		Sistema
1.	Introduce datos de la asignatura que desea buscar y marca la opción "Buscar" en la lista de asignaturas mostrados.	
2.		Verifica que los datos introducidos sean correctos
3-		Busca los datos de la asignatura y muestra una nueva lista con los datos de la asignatura encontrada. Finalizando así el Caso de Uso
Flujos alternos		
2a. "Datos incorrectos"		
Actor		Sistema
1.		No muestra ninguna opción en la lista de Asignaturas
Relaciones	CU Incluidos	No aplica
	CU Extendidos	No aplica
Requisitos no funcionales	No aplica	
Asuntos pendientes	No aplica	

CU 2. <Gestionar Balance de Carga>

Objetivo	Permite insertar, modificar, eliminar, listar y buscar los datos del Balance de Carga.
Actores	Profesor Principal de Año, Vicedecana.
Resumen	El caso de uso inicia cuando el Profesor Principal o la Vicedecana Inserta, modifica y elimina los datos del Balance de Carga.
Complejidad	Alta

Prioridad	Alta	
Precondiciones	Profesor Principal de Año o Vicedecana ya autenticado.	
Postcondiciones	Se registró, modificó o eliminó el Balance de Carga.	
Flujo de eventos		
Flujo básico < Gestionar Balance de Carga >		
Actor		Sistema
1.	Selecciona la opción "Balance de Carga".	
2.		Muestra en pantalla una lista con las opciones de Insertar datos, y Editar o Eliminar si ya existen datos insertados en la lista, además del listado de datos del balance de carga ya insertado.
3.	Desea Insertar, Editar o eliminar datos del Balance de Carga.	
4.		<p>El sistema despliega una pantalla con las opciones para la gestión del Balance de Carga.</p> <p>a) Para insertar datos al Balance de Carga ir a la sección "Insertar Datos"</p> <p>b) Para eliminar datos del Balance de Carga ir a la sección "Eliminar"</p> <p>c) Para modificar datos del Balance de Carga ir a la sección "Editar"</p>
Flujos alternos		
2a. Listado de Balance de Carga vacío.		
Actor		Sistema
1.		Se muestra en pantalla un listado vacío.
Sección 1: "Insertar Datos"		
Flujo básico < Insertar Datos >		
Actor		Sistema
1	Presiona la opción "Insertar Datos".	
2		Muestra un formulario con los siguientes campos a introducir:

		-Asignatura a Impartir -Tipo de Clases -Semana de Clases Y el botón "Guardar". Y el botón "Cancelar".
3	Introduce los datos del balance de carga en los campos y presiona la opción "Guardar" o "Cancelar".	
4		Verifica que los campos estén llenos.
5		Guarda los datos del balance de carga y muestra un mensaje "Datos registrados". Finalizando así el Caso de Uso.
Flujos alternos		
4a. Campos vacíos.		
Actor		Sistema
1.		Muestra un mensaje "Campos Requeridos".
Sección 2: "Editar Datos"		
Flujo básico < Editar Datos >		
Actor		Sistema
1	Presiona la opción "Editar" en el listado del Balance de Carga.	
2		Muestra un formulario con los siguientes campos a introducir: -Asignatura a Impartir -Tipo de Clases -Semana de Clases Y el botón "Editar". Y el botón "Cancelar".
3		Verifica que los campos estén llenos.

4		Actualiza los datos del Balance de Carga y muestra un mensaje "Balance de Carga Editado". Finalizando así el Caso de Uso.
Flujos alternos		
3a. Campos vacíos.		
Actor		Sistema
1.		Muestra un mensaje "Campos Requeridos".
Sección 3: "Eliminar Datos"		
Flujo básico <Eliminar Datos >		
Actor		Sistema
1.	Presiona el botón 'Eliminar' de la fila de datos que desee.	
2.		Muestra en pantalla un mensaje "Esta seguro que desea eliminar". -y el botón "Aceptar" -Y el botón "Cancelar"
3.	Presiona el botón "Aceptar "	
4.		Elimina los datos del Balance de Carga y muestra un mensaje "Datos eliminados". Finalizando así el Caso de Uso.
Flujos alternos		
2a. Marca botón 'Cancelar'		
Actor		Sistema
1.		Retornar al paso 2 en la sección "Balance de Carga"
Sección 4: "Buscar Datos"		
Flujo básico <Buscar Datos >		
Actor		Sistema
1.	Introduce datos del Balance de Carga que desea buscar y marca la opción "Buscar ".	
2.		Verifica que los datos introducidos sean correctos

3-		Busca los datos del listado de Balance de Carga y muestra una nueva lista con los datos encontrados. Finalizando así el Caso de Uso
Flujos alternos		
2a. "Datos incorrectos"		
Actor		Sistema
1.		No muestra ninguna opción en el Listado de Balance de Carga
Relaciones	CU Incluidos	No aplica
	CU Extendidos	No aplica
Requisitos no funcionales		No aplica
Asuntos pendientes		No aplica

CU 3. <Gestionar Afectaciones>

Objetivo	Permite insertar, modificar, eliminar, listar y buscar los datos de las posibles afectaciones.
Actores	Profesor Principal de Año, Vicedecana.
Resumen	El caso de uso inicia cuando el Profesor Principal de Año la Vicedecana inserta, modifica y elimina los datos de las posibles afectaciones.
Complejidad	Alta
Prioridad	Alta
Precondiciones	Profesor Principal de Año o Vicedecana ya autenticado.
Postcondiciones	Se registró, modificó o eliminó alguna Afectación.
Flujo de eventos	
Flujo básico < Gestionar Afectaciones >	
Actor	
Sistema	
1.	Selecciona la opción "Afectaciones".
2.	Muestra en pantalla una lista con las opciones de

		Insertar datos de las Afectaciones, y Editar o Eliminar si ya existen datos insertados en la lista, además del listado de afectaciones ya insertado.
3.	Desea Insertar, Editar o eliminar datos de las Afectaciones.	
4.		<p>El sistema despliega una pantalla con las opciones para la Gestión de las Afectaciones.</p> <p>a) Para insertar datos de alguna Afectación ir a la sección "Insertar Afectación"</p> <p>b) Para eliminar datos de alguna Afectación ir a la sección "Eliminar"</p> <p>c) Para modificar de alguna Afectación ir a la sección "Editar"</p>
Flujos alternos		
2a. Listado de Afectaciones vacío.		
Actor		Sistema
1.		Se muestra en pantalla un listado vacío.
Sección 1: "Insertar Afectación"		
Flujo básico < Insertar Afectación >		
Actor		Sistema
1	Presiona la opción "Insertar Afectación".	
2		<p>Muestra un formulario con los siguientes campos a introducir:</p> <ul style="list-style-type: none"> -Profesor Afectado -Semana -Día de la Semana Afectado -Turno de Clases -Año Docente <p>Y el botón "Guardar".</p>

		Y el botón "Cancelar".
3	Introduce los datos de las afectaciones en los campos y presiona la opción "Guardar" o "Cancelar".	
4		Verifica que los campos estén llenos.
5		Guarda los datos de las afectaciones y muestra un mensaje "Afectación Insertada". Finalizando así el Caso de Uso.
Flujos alternos		
4a. Campos vacíos.		
Actor		Sistema
1.		Muestra un mensaje "Campos Requeridos".
Sección 2: "Editar Datos"		
Flujo básico < Editar Datos >		
Actor		Sistema
1	Presiona la opción "Editar" en el listado de Afectaciones.	
2		Muestra un formulario con los siguientes campos a introducir: -Profesor Afectado -Semana -Día de la Semana Afectado -Turno de Clases -Año Docente Y el botón "Guardar". Y el botón "Cancelar".
3		Verifica que los campos estén llenos.
4		Actualiza los datos del listado de Afectaciones y muestra un mensaje "Afectación Editada".

		Finalizando así el Caso de Uso.
Flujos alternos		
3a. Campos vacíos.		
Actor		Sistema
1.		Muestra un mensaje "Campos Requeridos".
Sección 3: "Eliminar Datos"		
Flujo básico <Eliminar Datos >		
Actor		Sistema
1.	Presiona el botón 'Eliminar' en el listado de Afectaciones.	
2.		Muestra en pantalla un mensaje "Esta seguro que desea eliminar". -y el botón "Aceptar" -Y el botón "Cancelar"
3.	Presiona el botón "Aceptar "	
4.		Elimina los datos del listado de Afectaciones y muestra un mensaje "Datos eliminados". Finalizando así el Caso de Uso.
Flujos alternos		
2a. Marca botón Cancelar		
Actor		Sistema
1.		Retornar al paso 2 en la sección "Afectaciones"
Sección 4: "Buscar Datos"		
Flujo básico <Buscar Datos >		
Actor		Sistema
1.	Introduce datos de la Afectación que desea buscar y marca la opción "Buscar ".	
2.		Verifica que los datos introducidos sean correctos
3-		Busca los datos del listado de Afectaciones y muestra una nueva lista con los datos encontrados.

		Finalizando así el Caso de Uso
Flujos alternos		
2a. "Datos incorrectos"		
Actor		Sistema
1.		No muestra ninguna opción en el Listado de Afectaciones.
Relaciones	CU Incluidos	No aplica
	CU Extendidos	No aplica
Requisitos no funcionales		No aplica
Asuntos pendientes		No aplica

2.4 Diseño de la propuesta de solución para la confección del horario docente.

El análisis y diseño del módulo permite conocer las principales propiedades y limitantes que el mismo debe poseer, con el objetivo primordial de otorgarle al cliente la mayor satisfacción posible. En el presente epígrafe se muestran algunos de los Diagramas de Clases de Diseño basados en Estereotipos Web, así como varios Diagramas de Secuencias.

2.4.1 Diagramas de clases de diseño con estereotipos web

Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema concreto al modelar sus clases, atributos, operaciones y relaciones entre objetos.

Los diagramas de clase con estereotipos web pueden utilizarse cuando se desarrolla diagramas de caso de uso del sistema, como en la presente investigación. Los mismos, describen gráficamente las especificaciones descritas en el sub-epígrafe 2.3.4. Entre los beneficios más conocidos de los diagramas de clases, se pueden encontrar que permiten ilustrar modelos de datos para sistemas de información, además, permite comprender mejor la visión general de los esquemas de una aplicación.

En la investigación se generaron un total de 7 diagramas de clases, por lo que a continuación se muestra tres de ellos:

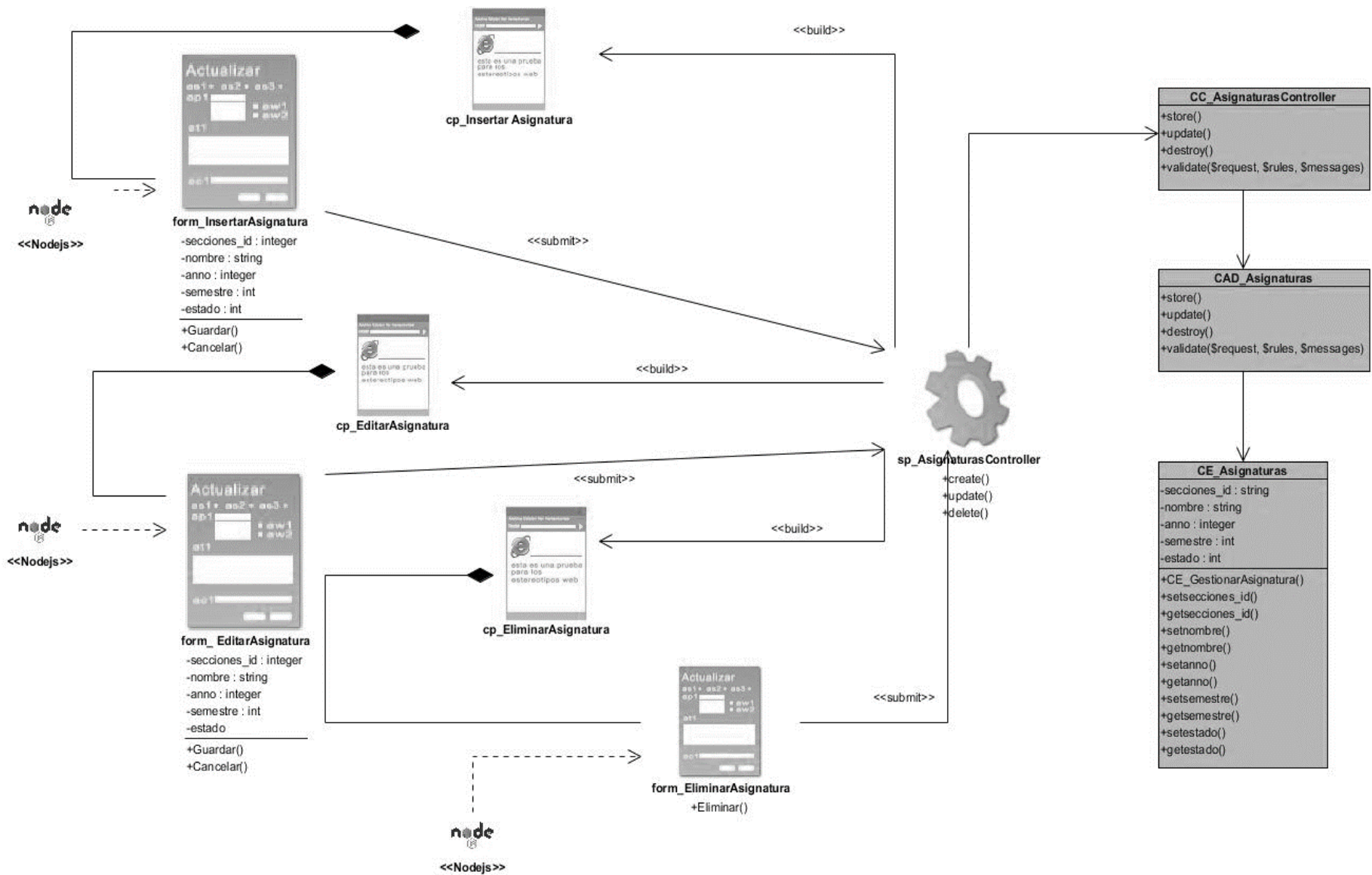


Figura 3. Diagrama de Clases con Estereotipos Web (Gestionar Asignaturas).

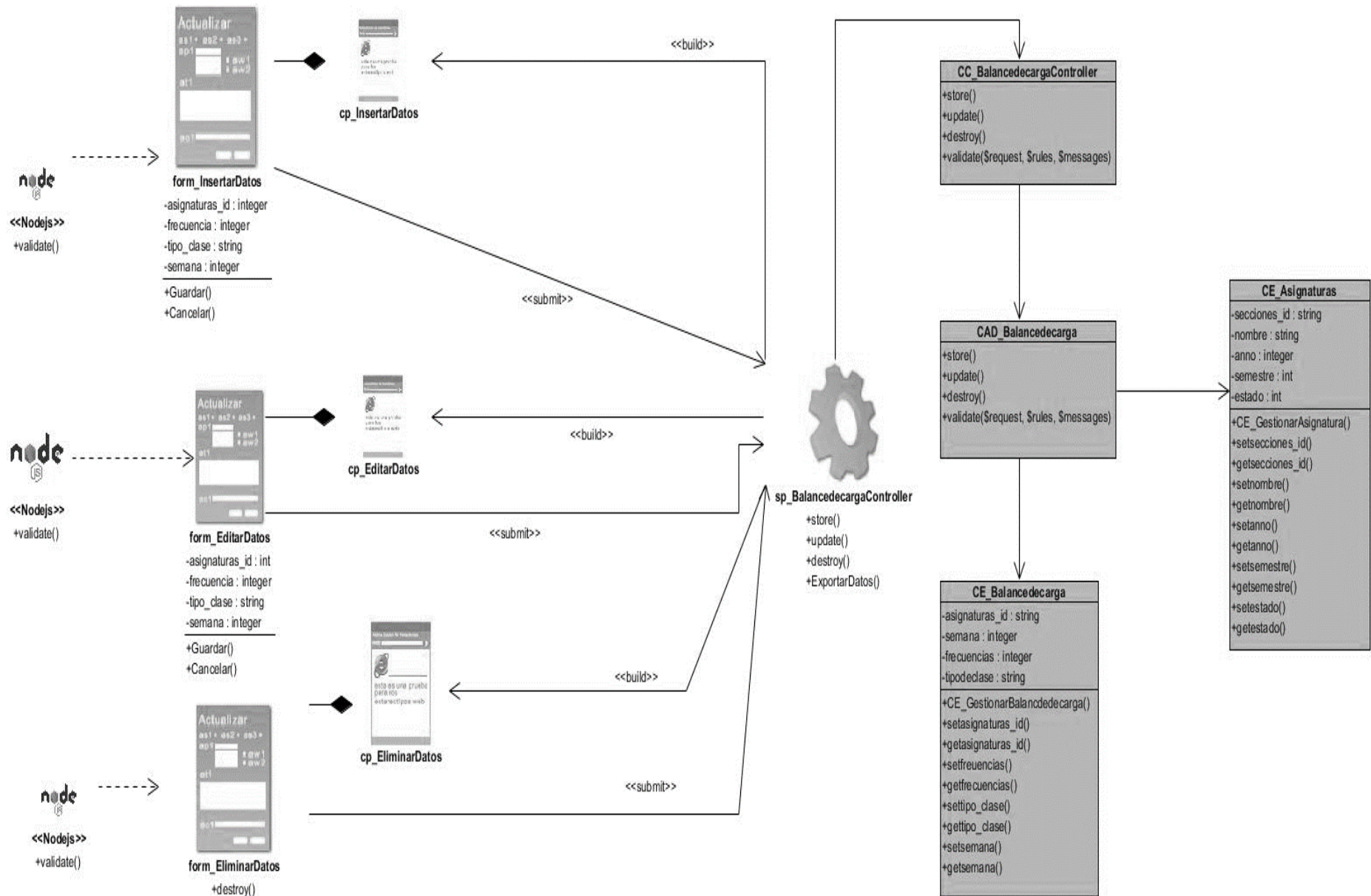


Figura 4. Diagrama de Clases con Estereotipos Web (Gestionar Balance de Carga).

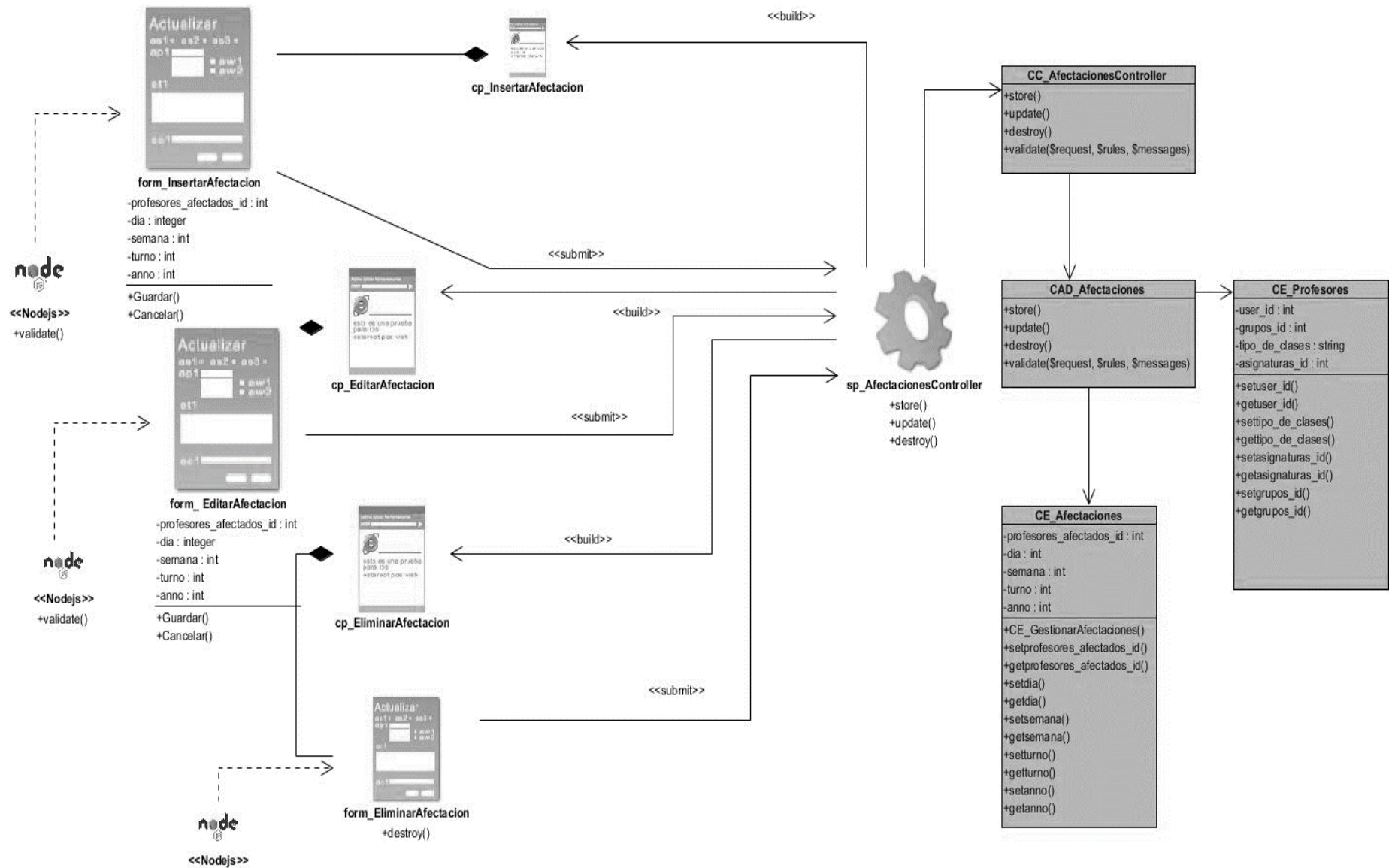


Figura 5. Diagrama de Clases con Estereotipos Web (Gestionar Afectaciones).

2.4.2 Diagramas de secuencias

Un diagrama de secuencia es un tipo de diagrama de interacción, que describe cómo, y en qué orden un grupo de objetos funcionan en conjunto. La mayoría de los profesionales de negocios usan estos diagramas para comprender los requisitos de un sistema nuevo o documentar un proceso existente.

Entre los principales beneficios que se puede obtener al realizar diagramas de secuencias están:

- Representan los detalles de un caso de uso en UML.
- Permiten modelar la lógica de una operación, una función o procedimientos sofisticados.
- Da paso a planificar y comprender la funcionalidad detallada de un escenario actual o futuro.

Para la presente investigación se generaron un total de 21 diagramas de secuencias, a continuación, se muestran 3 de ellos:

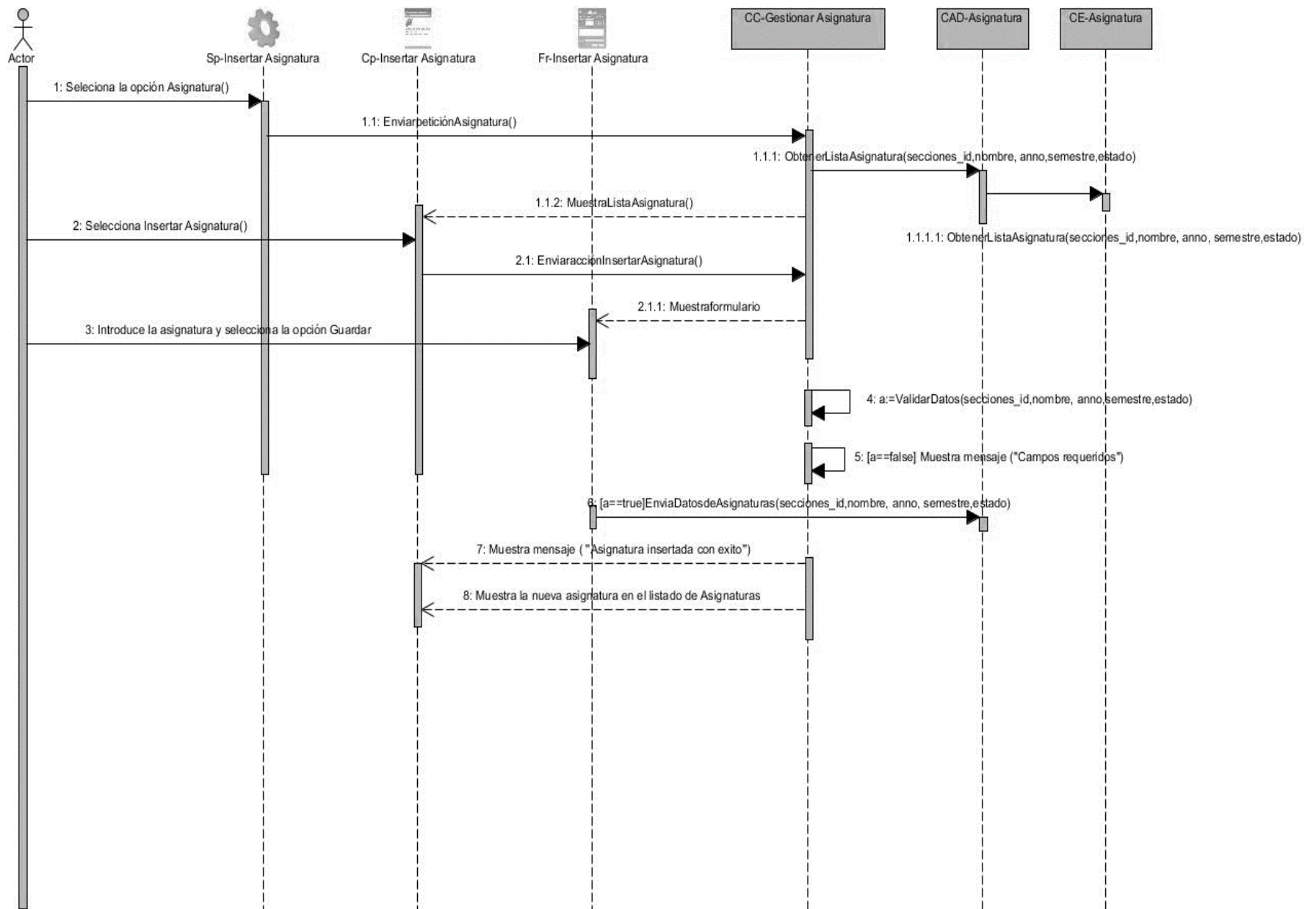


Figura 6. Diagrama de Secuencia (Insertar Asignatura).

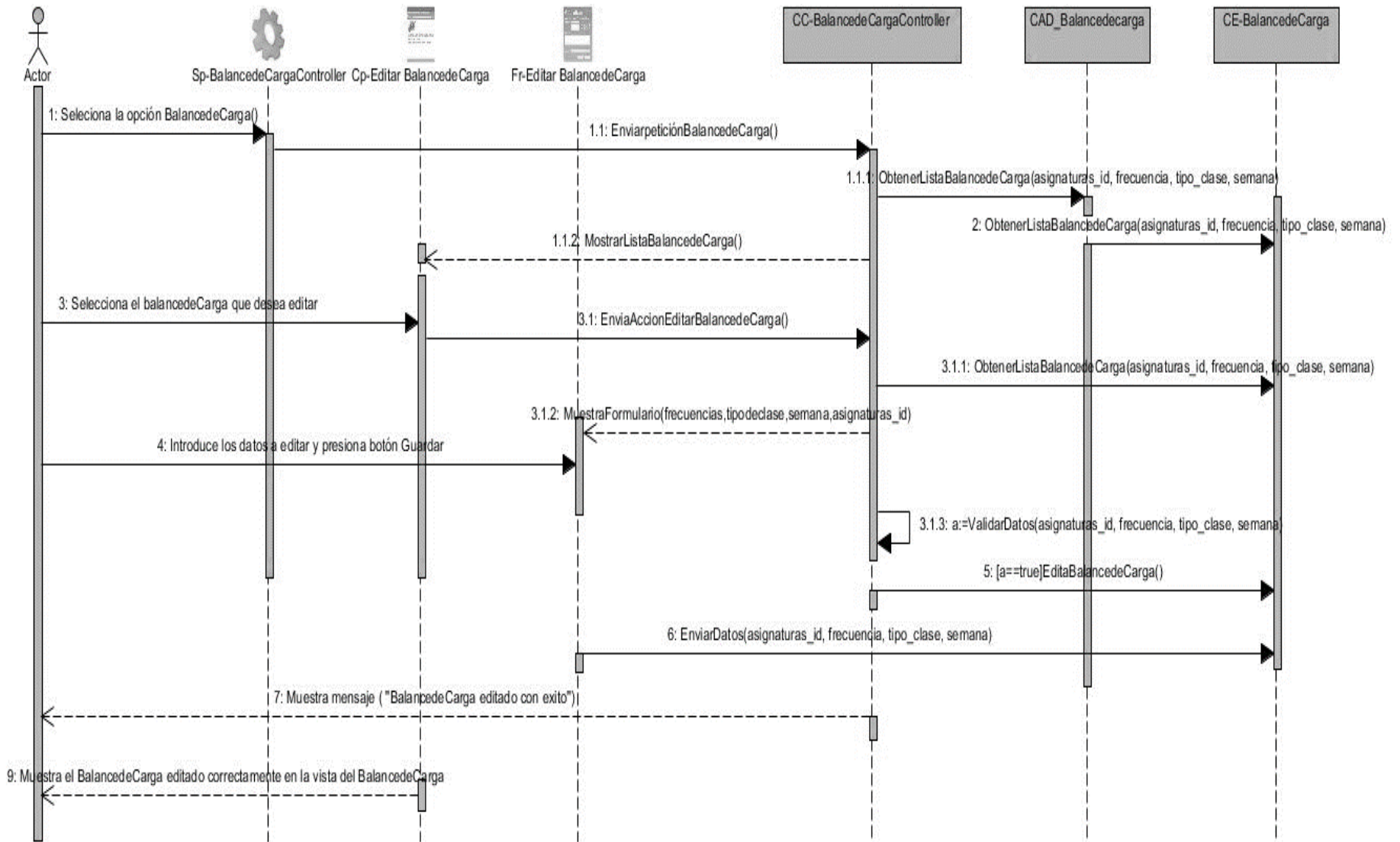


Figura 7. Diagrama de Secuencia (Editar Balance de Carga)

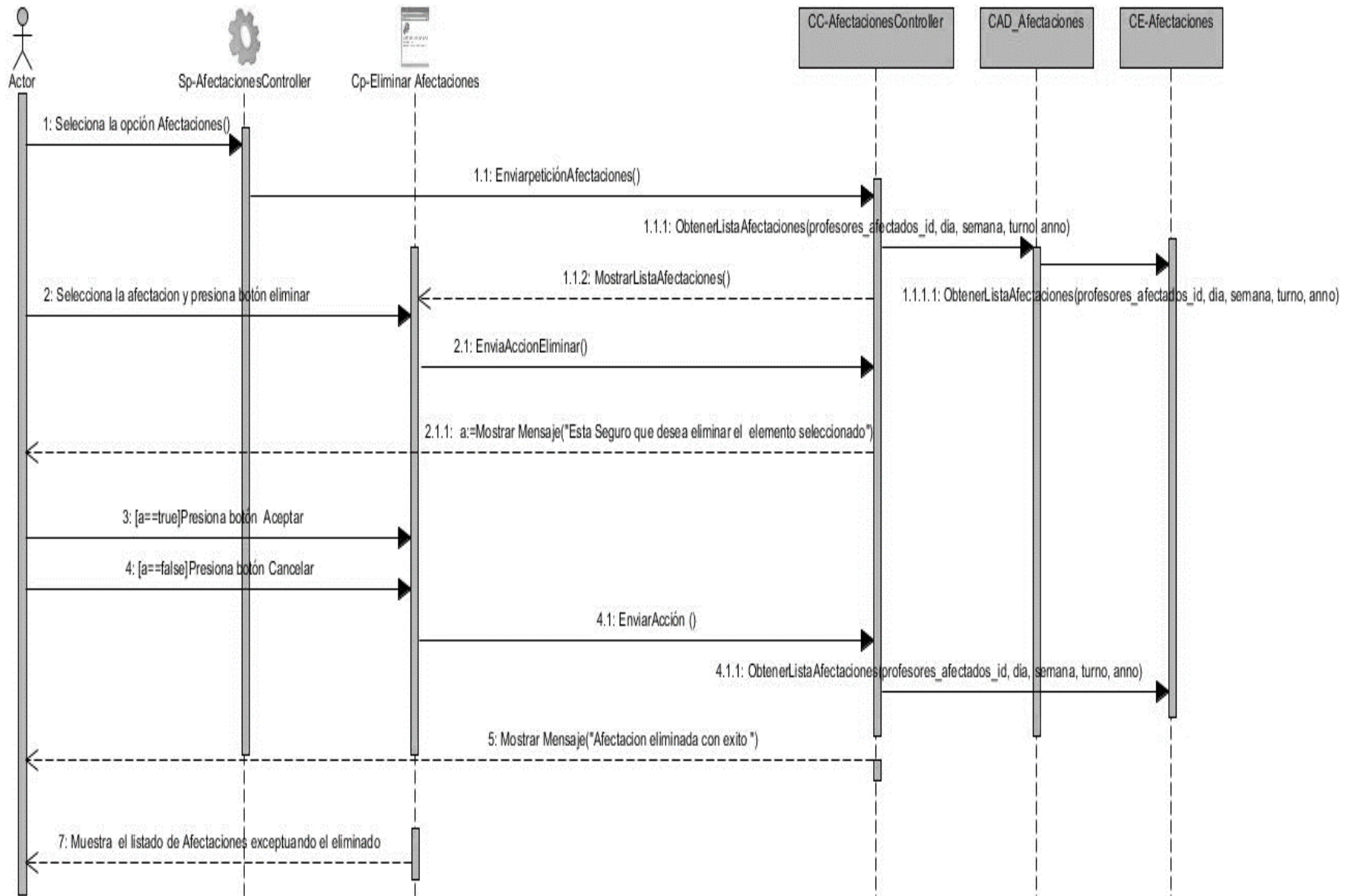


Figura 8. Diagrama de Secuencia (Eliminar Afectaciones).

2.4.3 Modelo de datos

El Modelado de Datos es el proceso que generalmente se utiliza para documentar un diseño de sistema de software complejo como un diagrama de fácil comprensión, usando básicamente texto y símbolos para representar la forma en que los datos necesitan fluir, que puede ser usado como un diagrama de flujo que ilustra las relaciones entre los datos. Tradicionalmente, los modelos de datos se han construido durante las fases de análisis y diseño de un proyecto, con el principal objetivo de asegurar que los requisitos para la nueva aplicación a construir se comprendan completamente (Hamilton, 2015).

En la siguiente Figura 9. Modelo de Datos, se puede observar la cantidad de tablas, así como las relaciones existentes entre las mismas. Ejemplo claro de lo anterior mencionado es la tabla profesores, la cual está relacionada a través de una relación de uno a uno con la tabla users y la tabla asignaturas. Además, existe una relación de mucho a mucho entre tres tablas (profesores, grupos, asignaturas) creando una nueva tabla correspondiente a dicha relación que es la tabla planificacions. Existe además la tabla balancedecarga, la cual está relacionada mediante una relación de uno a mucho con la tabla asignaturas, a su vez se puede observar la tabla secciones que se encuentra relacionada con la tabla asignaturas usando una relación de uno a mucho. También se encuentra representado en el modelo la tabla afectaciones que está relacionada con la tabla profesores utilizando una relación de uno a muchos, así como la tabla local, la cual se encuentra relacionada con la tabla disponibilidad y tipo_de_locales a través de una relación de uno a mucho. Por último, se puede apreciar la tabla asignaciones, que está relacionada a través de una relación de uno a muchos con las tablas disponibilidad y planificacions.

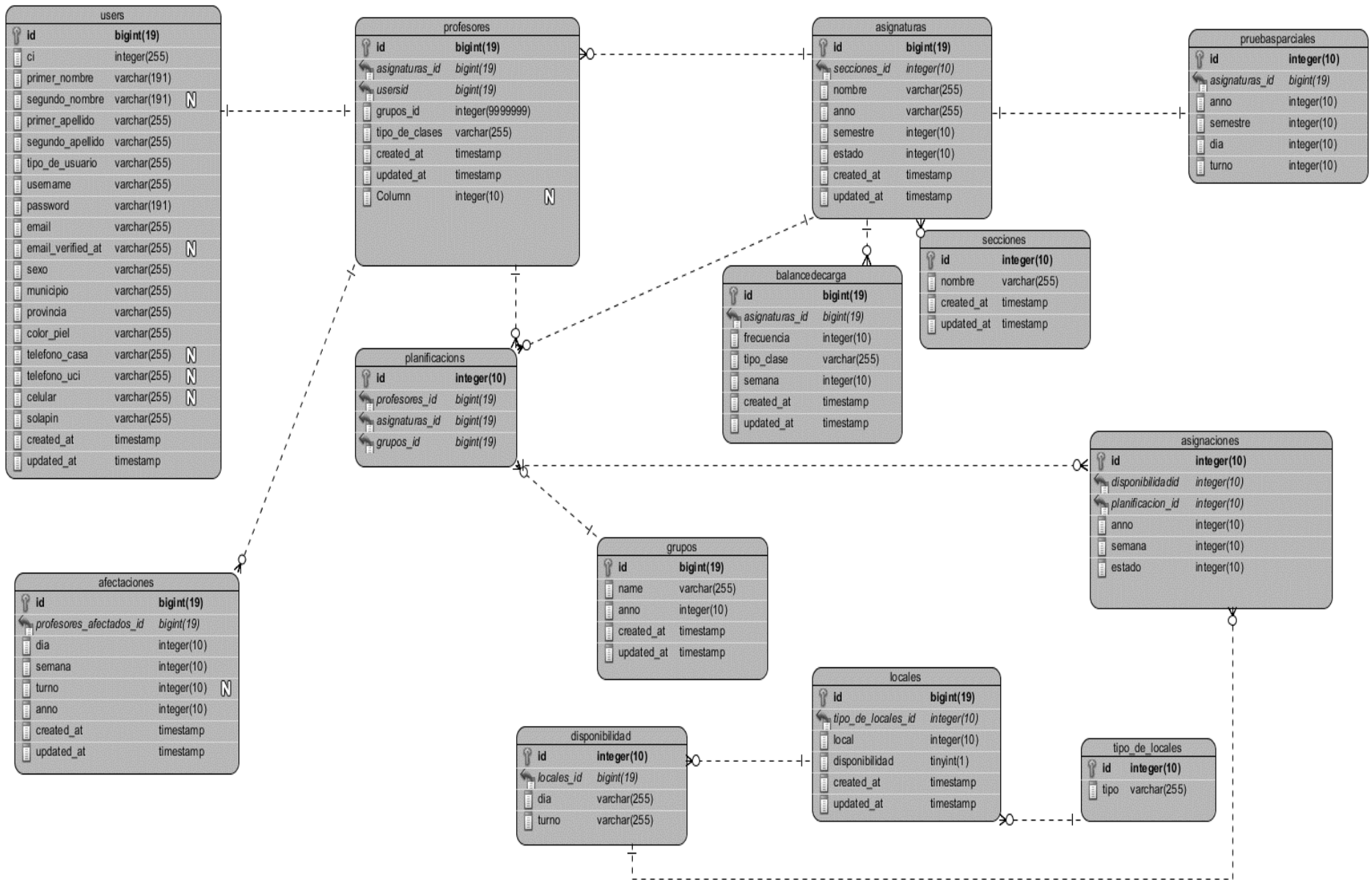


Figura 9. Modelo de Datos

2.4.4 Arquitectura

El Modelo-Vista-Controlador, surge con el objetivo de reducir el esfuerzo de programación, necesario en la implementación de sistemas múltiples y sincronizados de los mismos datos, a partir de estandarizar el diseño de las aplicaciones. Es un paradigma que divide las partes que conforman una aplicación en el Modelo, las Vistas y los Controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo (Romero, 2012).

En la Figura 10. Modelo-Vista-Controlador, se explica el funcionamiento de dicho estilo arquitectónico, y se presenta además la serie de pasos lógicos a modo de ejemplo, de cómo se utilizaría dicha arquitectura en la presente investigación:

1. El usuario envía una petición desde la Vista Asignaturas.
2. La vista asignatura interactúa con el controlador AsignaturasController enviando los datos que solicite el cliente en la vista.
3. El controlador AsignaturasController envía al modelo Asignaturas los datos recibidos de la vista para obtener una respuesta.
4. El modelo Asignaturas envía la respuesta al controlador AsignaturasController.
5. El controlador AsignaturasController envía la respuesta recibida del modelo Asignaturas hacia la vista Asignaturas.
6. La vista Asignaturas envía la respuesta de los datos recibidos del controlador al usuario.

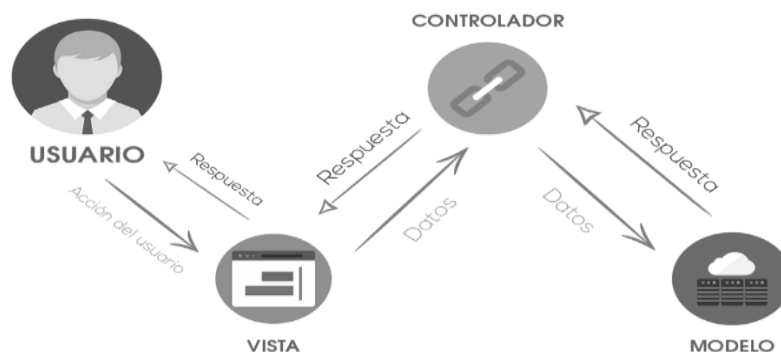


Figura 10. Modelo-Vista-Controlador

Luego de analizar el comportamiento principal de la arquitectura Modelo-Vista-Controlador, podemos resumir de manera más específica las principales funcionalidades de cada componente esencial de la arquitectura:

- **Modelo:** Se encarga de los datos, generalmente (pero no obligatoriamente) consultando la base de datos, además, realiza actualizaciones, consultas, búsquedas, entre otras funciones.
- **Vistas:** Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica va aquí. Su principal funcionalidad es encargarse de cómo se verán los datos.
- **Controlador:** Se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista básicamente.

2.4.5 Patrones de diseño

Los patrones de diseño son ese mecanismo que puede ayudar a aliviar tanto el acometido de nuevos desarrollos como la dificultad en la toma de decisiones para la mejora de sistemas ya existentes, una vez que puedan integrarse de forma eficiente con los métodos de diseño y sean accesibles de manera efectiva a los diferentes integrantes del equipo de desarrollo. Además, los mismos se perfilan como una técnica que favorece la reutilización de conocimiento de diseño, ya que hacen posible la expresión de una solución ya probada a un problema recurrente (Díaz, 2005).

Patrones Generales de Software para la Asignación de Responsabilidades (GRASP):

Estos patrones se utilizan en la programación orientada a objetos, con lo que el punto de partida para aplicarlos siempre será usar POO. Cómo definir estos objetos/clases será la aplicación de cada patrón.

- **Experto:** El patrón Experto se usa con frecuencia en la asignación de responsabilidades, es un principio básico que se utiliza continuamente en el diseño de objetos. El denominado experto expresa por lo general la intuición común de que los objetos realizan las acciones relacionadas con la información que poseen. En la Figura 11, se puede observar la clase GestionarAsignatura, experta en información relacionada con las asignaturas.



Figura 11. Empleo del Patrón Experto

- **Bajo Acoplamiento:** El bajo acoplamiento entre objetos se refiere a cuánta relación tienen los objetos entre sí. En la presente investigación, por ejemplo, al mostrar el nombre de las asignaturas correspondientes a un año docente en el balance de carga, se tiene que recorrer la lista de asignaturas existentes, por lo que entonces aumentamos el acoplamiento entre Asignaturas y Balance de Carga. La solución que brinda el patrón para evitar que aumente el acoplamiento, y por ende se mantenga bajo, es asignar las responsabilidades de forma tal que las clases se comuniquen con el menor de ellas posibles, que es precisamente lo que se realiza en la investigación, por ejemplo, donde se asigna la menor responsabilidad posible entre las clases.
- **Alta Cohesión:** Este patrón conduce a la asignación de responsabilidades relacionadas con un número relativamente pequeño. Para ello propone asignar a las clases las responsabilidades que trabajen sobre una misma área de la aplicación y que no tengan mucha complejidad. En la investigación, el patrón se evidencia mayormente en cada clase que realiza una labor única dentro del sistema y colabora con las otras para llevar a cabo tareas específicas, por lo general en las clases que forman parte de la capa de acceso a datos. Un ejemplo claro de esto puede ser la clase Gestionar Afectaciones, la cual no realiza un gran número de métodos, además de que no suele realizar mucho trabajo, y colabora también con la clase Profesor para compartir el esfuerzo.

CE_Afectaciones
-profesores_afectados_id : int
-dia : int
-semana : int
-turno : int
-anno : int
+CE_GestionarAfectaciones()
+setprofesores_afectados_id()
+getprofesores_afectados_id()
+setdia()
+getdia()
+setsemana()
+getsemana()
+setturno()
+getturno()
+setanno()
+getanno()

Figura 12. Empleo del Patrón Alta Cohesión.

- **Controlador:** Este patrón define quién debería encargarse de atender un evento del sistema. La solución consiste en asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase controladora. En la investigación se ve reflejado en todas las clases “Controllers”, ya que contienen todos los métodos necesarios para controlar las actividades que se soliciten. Ejemplo específico de lo antes mencionado puede ser la clase AsignaturasController, que se encarga de atender todas las solicitudes de eventos necesarios relacionados con la clase Asignatura.
- **Creador:** Se basa principalmente en la asignación de responsabilidades relacionadas con la creación de objetos. Es el encargado de crear las instancias de los objetos para cada una de las funcionalidades de la aplicación. El patrón se evidencia en las clases controladoras que, para cada uno de las funcionalidades del módulo, se encargan de crear las instancias de los objetos que manejan, favoreciendo así la reutilización y el bajo acoplamiento, un ejemplo puede ser el controlador AsignaturasController.

Patrones Gang of Four (GOF)

Los patrones GoF, conocidos como el grupo de los cuatro, se clasifican en tres grupos dependiendo de sus propósitos, Creacional, Estructural y Comportamiento. Estos describen soluciones simples a

problemas específicos en el diseño de software orientado a objetos y otros ámbitos referentes al diseño de interfaces (Hamilton, 2015).

- **Factory:** El patrón de diseño Factory nos ayuda a crear instancias de otras clases, con la finalidad de ocultar la complejidad que se requiere para crearlas. En la investigación el patrón se puede evidenciar en la función `devolver_locales()`, en la cual uno de los parámetros que recibe es la sección en la que se impartirá la clase, con la cual habría que realizar varias consultas y conexiones a la base de datos (una por cada caso), algo que se evita al utilizar dicho patrón, pues con el mismo se realiza la consulta y la conexión una sola vez.

```
$result = $conn->query($sql);  
  
if ($result->num_rows > 0) {  
    while ($row = $result->fetch_assoc()) {  
        array_push($locales_disponibles, array("id" => intval($row['id']), "local_id" => intval($row['local_id']), "dia" => intval($row['dia']), "turno"  
    )  
    }  
}  
  
return $locales_disponibles;
```

Figura 13. Uso del patrón de diseño GOF (Factory)

- **Decorador:** Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (González Rodríguez, 2017). Se puede observar el uso de este patrón en la propuesta de solución en todas las vistas, ya que todas heredan de una plantilla `AdminLTE`, la cual se comporta como la plantilla principal del proyecto. En la siguiente Figura se puede observar el uso de dicho patrón:

```
@extends('adminlte::page')
```

Figura 14. Uso del Patrón Decorador

2.4.6 Diagrama de despliegue

Un Diagrama de Despliegue no es más que un tipo de diagrama UML en el cual generalmente se muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, por mencionar algunos. Los mismos por lo general ayudan a modelar la topología de hardware de

un sistema en comparación con otros tipos de diagramas UML, que en su mayoría esbozan los componentes lógicos de un sistema.

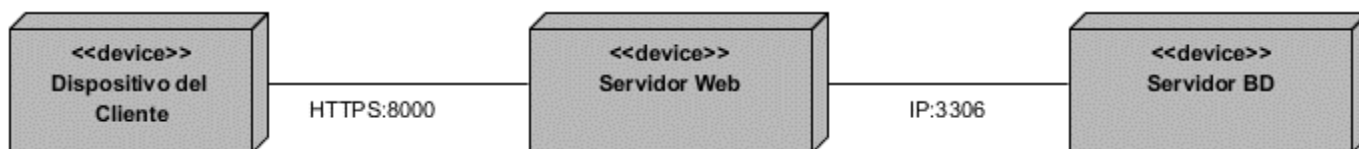


Figura 15. Diagrama de Despliegue.

Servidor Web: Es donde se puede observar el código fuente de la aplicación, brinda a los usuarios las interfaces necesarias durante todos los procesos del negocio. Esta estación se comunica con el servidor de base de datos MySQL, realizando la comunicación a través del protocolo IP:3306. Debe poseer un disco duro con capacidad superior a los 80 GB, contar con al menos con una tarjeta RAM de 2 GB de capacidad, además, debe poseer una tarjeta de red con velocidad aproximada a los 100 Mbps, debe tener incorporado un servidor Web Apache con PHP 8.0.3.

Dispositivo del cliente: Contiene la estación de trabajo del cliente que necesita un navegador web para conectarse con el servidor, el cual utiliza para conectarse el protocolo de comunicación HTTPS. El cliente debe poseer un navegador, principalmente Chrome o Mozilla Firefox, debe usar como sistema operativo Linux o Windows, además debe contar con una tarjeta de red incorporada con velocidad cercana a los 100 Mbps. También debe tener incluida una tarjeta RAM de 2 GB como mínimo y un procesador de 2 GHZ como mínimo.

Servidor de BD: Este nodo es el encargado del almacenamiento de los datos registrados en la aplicación. Se comunica con el servidor de aplicaciones a través del protocolo IP, permitiendo el acceso a la información que tanto el usuario como la aplicación pueden manipular. El servidor debe tener una capacidad mínima de 20 GB, además de que debe contar una tarjeta de red con velocidad aproximada a los 10 Mbps, y debe contar con una tarjeta RAM con capacidad mínima de 2 GB. Es importante que cuente además con MySQL en alguna versión superior a la 8.0 como gestor de base de datos.

CONCLUSIONES PARCIALES DEL CAPÍTULO

En el presente capítulo luego de realizado el análisis y diseño de la propuesta de solución, junto a la confección de los distintos artefactos que propone la metodología AUP-UCI, se puede concluir que el análisis y estudio de los sistemas homólogos nacionales, así como las entrevistas realizadas a distintas personas dentro de la Facultad 1, permitieron la obtención de los requisitos funcionales y no funcionales del Módulo de Generación del Horario Docente de la Facultad 1. Además, la construcción de los diagramas de clases con estereotipos web y los diagramas de secuencias, proporcionan una perspectiva en cuanto a la estructura y composición del módulo. También con el desarrollo de todos los artefactos que se necesitan en el módulo, se podrá construir una documentación de la solución, lo que servirá de apoyo para futuros mantenimientos del módulo.

CAPÍTULO 3: IMPLEMENTACIÓN Y VALIDACIÓN DEL MÓDULO DE GENERACIÓN DE HORARIOS DOCENTES PARA EL SISTEMA DE TRABAJO EDUCATIVO EN LA FACULTAD 1.

En la realización del actual capítulo se describe el proceso de implementación y pruebas del Módulo de Generación de Horarios Docentes para el Sistema de Trabajo de la Facultad 1

3.1 Introducción

Según (SOMMERVILLE, 2005), la implementación de un sistema o módulo constituye una de las fases más importantes del desarrollo de software. En ella se toman como punto de partida los resultados obtenidos en el diseño, implementándose el sistema o módulo en términos de componentes como ficheros de código binario, código fuente, scripts y ejecutables.

Todo proceso de implementación debe ir acompañado de alguna actividad que garantice la calidad. Las pruebas de validación constituyen una base para garantizar la aceptación favorable de una aplicación informática por parte del usuario. Con la realización de las mismas se pretende encontrar y documentar los errores que tiene un sistema, validar los requisitos y comprobar que estos fueron implementados correctamente. Este capítulo tiene como objetivo, documentar los resultados de las fases de implementación del módulo y de la estrategia de pruebas desarrollada, así como realizar la validación científica de la propuesta de solución mediante la técnica de ladov.

3.2 Diagrama de Componentes

Los diagramas de componentes de manera general muestran un conjunto de componentes y sus relaciones de manera gráfica a través del uso de nodos y arcos entre estos, además de que muestran la organización y las dependencias existentes entre un conjunto de componentes (Sánchez, 2011).

A continuación, se muestran tres figuras, en las cuales se representan los Diagramas de Componentes correspondientes a los casos de uso descritos en el capítulo anterior. En la Figura 16, se puede observar el diagrama de componentes correspondiente al CRUD Gestionar Asignaturas, en la Figura 17, el CRUD correspondiente al Gestionar Afectaciones, y en la Figura 18 el CRUD relacionado con la gestión del Balance de Carga, los anteriormente mencionados se encuentra elaborados y organizados de acuerdo con el Modelo-Vista-Controlador utilizado por el Framework de desarrollo de PHP llamado Laravel.

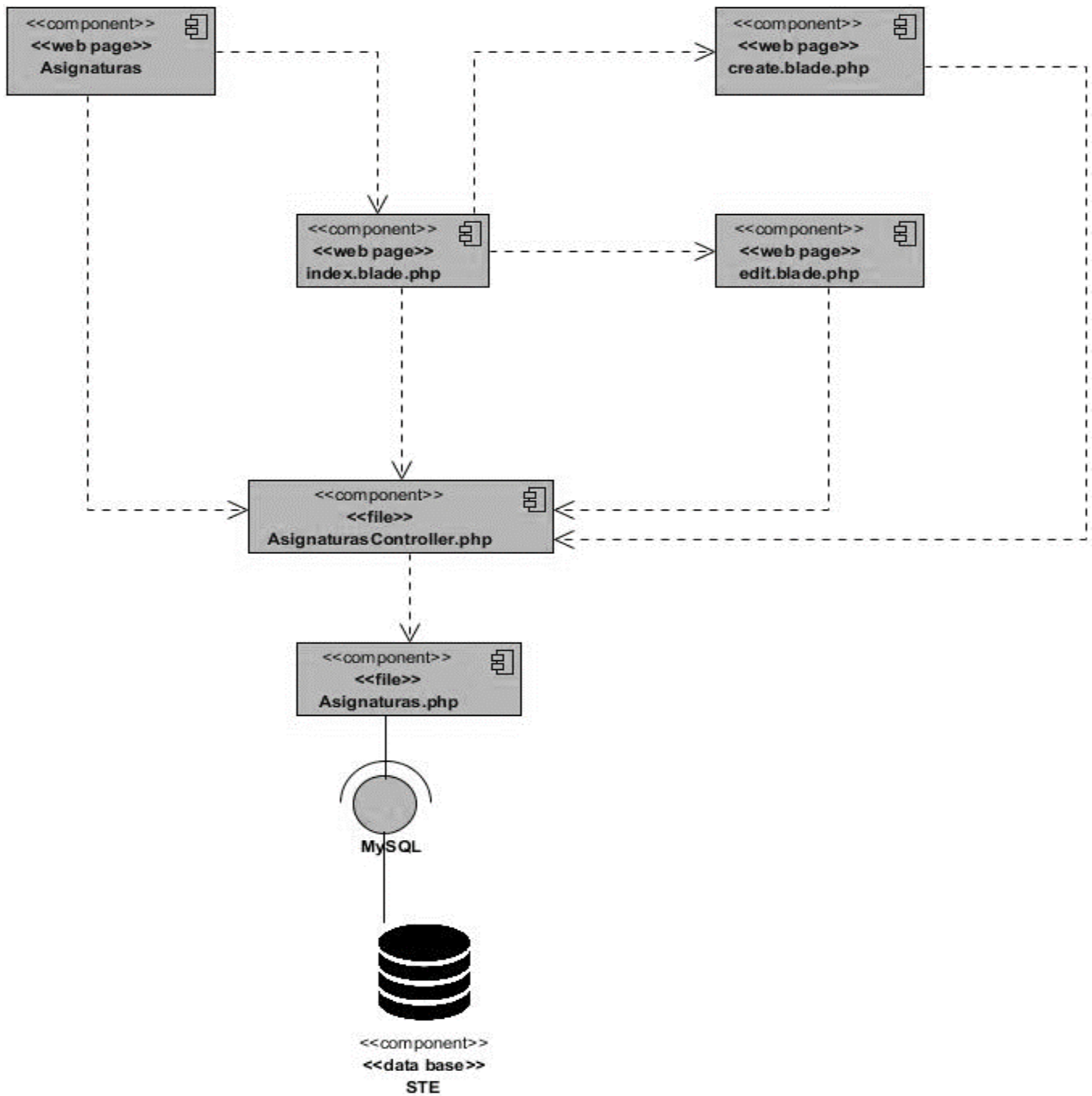


Figura 16. Diagrama de Componentes Gestionar Asignaturas.

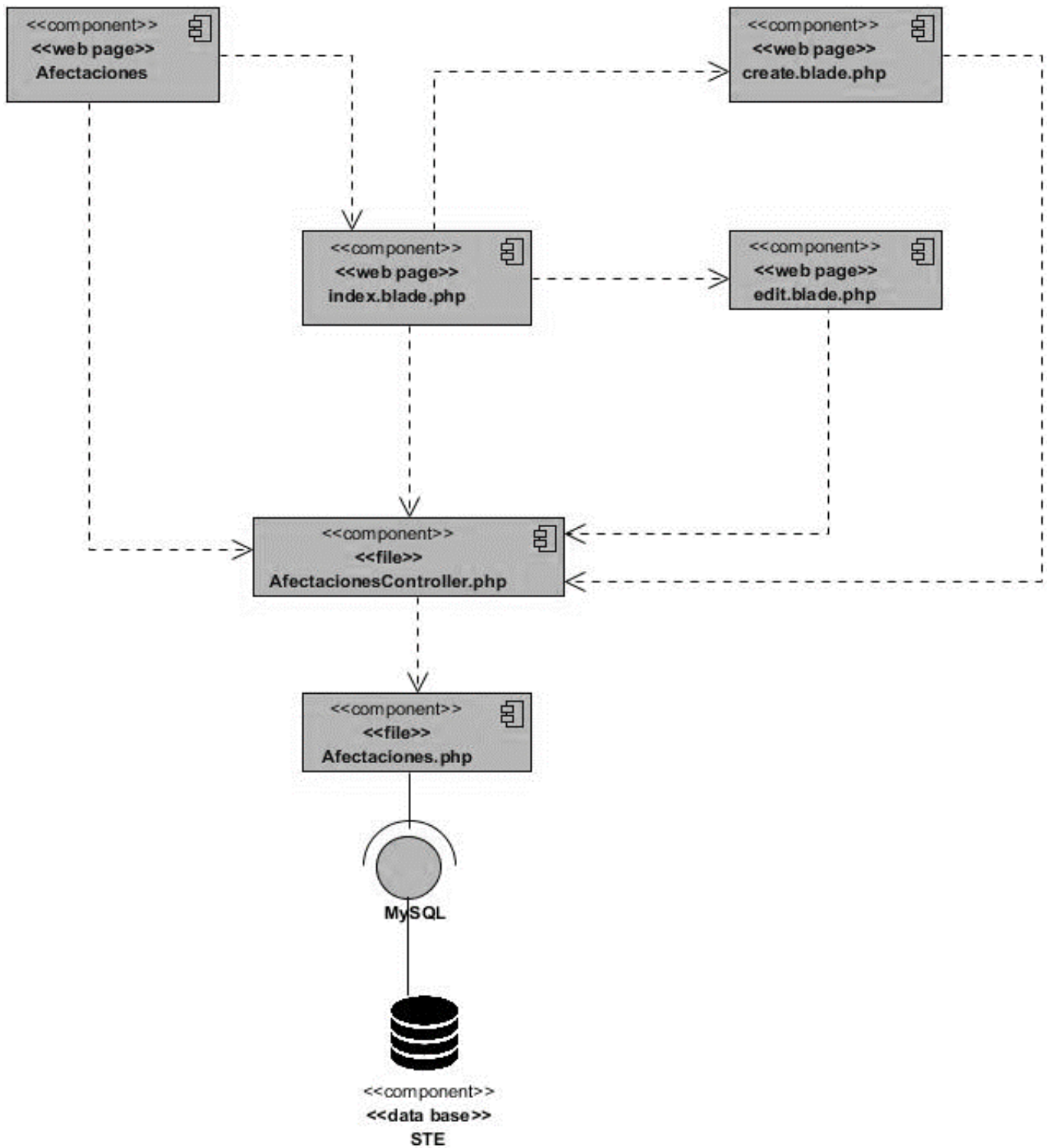


Figura 17. Diagrama de Componentes Gestionar Afectaciones.

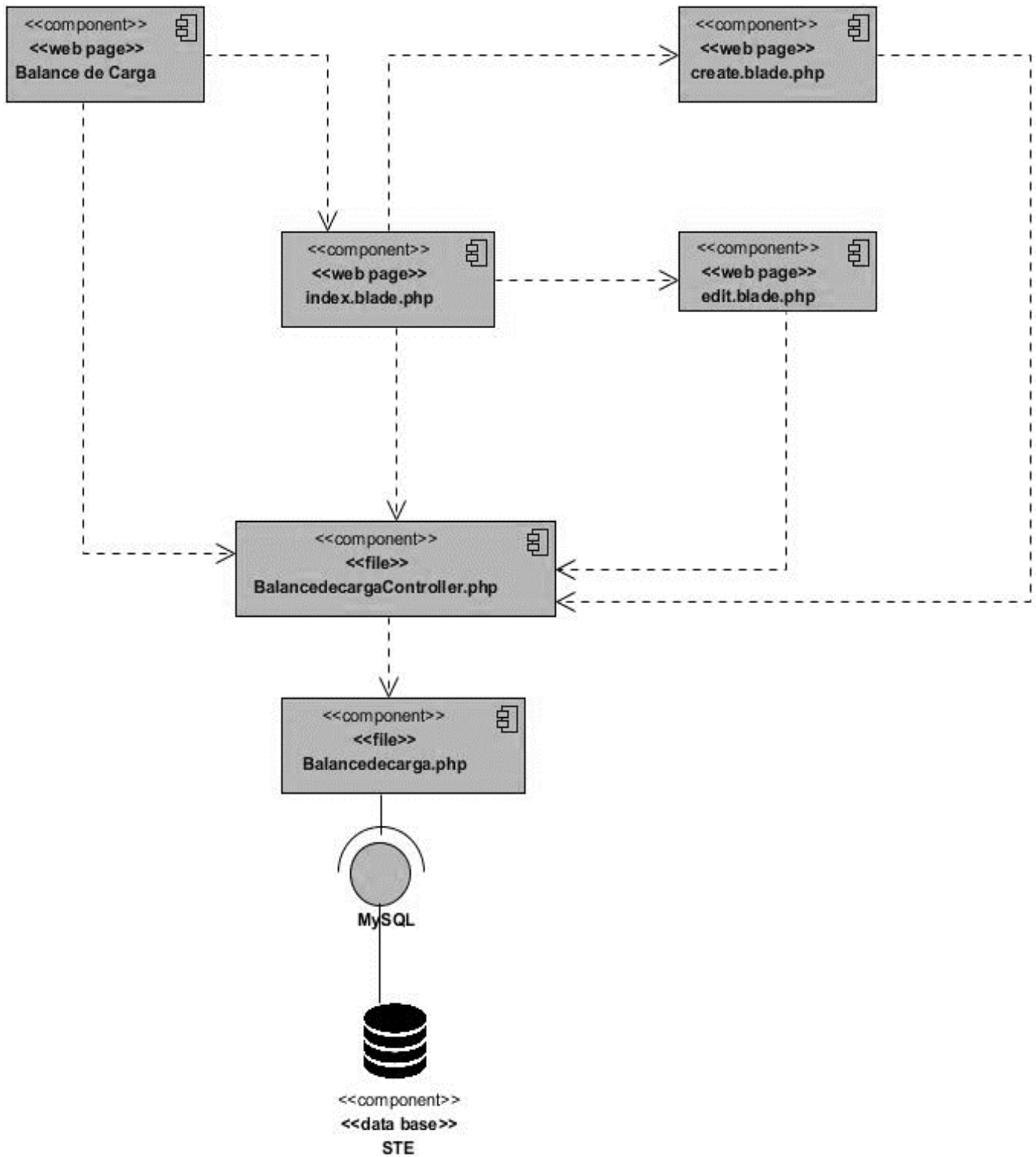


Figura 18. Diagrama de Componentes Gestionar Balance de Carga.

Con el principal objetivo de comprender los anteriores Diagramas de Componentes mostrados, se describen a continuación en varias tablas los elementos principales que forman parte de los mismos:

Tabla 3. Descripción del Diagrama de Componentes Gestionar Asignaturas.

Componentes		Descripción
Modelo	Asignaturas.php	Clase entidad que almacena los datos correspondientes a las Asignaturas insertadas en la base de datos.
Vistas	Asignaturas	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las asignaturas.
	index.blade.php	Este componente se encarga de mostrar un listado de las asignaturas insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar las asignaturas.
	create.blade.php	Este componente se encarga de registrar una nueva asignatura en la base de datos con los datos necesarios de las mismas.
	edit.blade.php	Este componente se encarga de editar una asignatura en la base de datos con los datos necesarios de las mismas.
Controlador	AsignaturasController.php	Este componente se encarga de realizar todos los métodos relacionados a la gestión de las asignaturas en el módulo.

Tabla 3.1. Descripción del Diagrama de Componentes Gestionar Afectaciones.

Componentes		Descripción
Modelo	Afectaciones.php	Clase entidad que almacena los datos correspondientes a las Afectaciones insertadas en la base de datos.

Vistas	Afectaciones	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestionan las afectaciones.
	index.blade.php	Este componente se encarga de mostrar un listado de las afectaciones insertadas en la base de datos, así como mostrar las opciones necesarias para gestionar las mismas.
	create.blade.php	Este componente se encarga de registrar una nueva afectación en la base de datos con los datos.
	edit.blade.php	Este componente se encarga de editar una afectación en la base de datos.
Controlador	AfectacionesController.php	Este componente se encarga de realizar todos los métodos relacionados a la gestión de las afectaciones en el módulo.

Tabla 3.2. Descripción del Diagrama de Componentes Gestionar Balance de Carga.

Componentes		Descripción
Modelo	Balancedecarga.php	Clase entidad que almacena los datos correspondientes al Balance de Carga insertados en la base de datos.
Vistas	Balance de Carga	Este componente se encarga de redireccionar desde la página principal del sistema hacia la vista donde se gestiona el balance de carga.
	index.blade.php	Este componente se encarga de mostrar un listado del balance de carga ya insertado en la base de datos, así como mostrar las opciones necesarias para gestionar el mismo.
	create.blade.php	Este componente se encarga de registrar nuevos datos del balance de carga en la base de datos.
	edit.blade.php	Este componente se encarga de editar los datos de alguna fila específica del balance de carga en la base de datos.

Controlador	BalancedecargaController.php	Este componente se encarga de realizar todos los métodos relacionados a la gestión del balance de carga en el módulo.
-------------	------------------------------	---

3.3 Estándares de codificación

Un estándar de codificación comprende generalmente todos los aspectos de la generación base del código. Los programadores deben implementar un estándar prudente y lógico, lo que debe tender siempre a que el código fuente sea completo y se refleje de manera armoniosa, como si un solo un solo programador hubiera escrito todo el código de una sola vez (Herrera, 2017).

A continuación, se pretende establecer o definir los estándares de codificación que se estarán utilizando durante la implementación del módulo que se trabaja en la presente investigación:

Tabla 3.3. Estándares de Codificación.

Tipo de Estándar	Descripción
Nomenclatura de las Clases	Los nombres de las clases empiezan con mayúsculas y el resto del nombre con minúscula, en caso que sea un nombre compuesto la segunda palabra también comenzará con mayúscula. Ejemplo de esto es la clase Asignaturas, donde se puede observar que la clase posee un solo nombre y comienza con mayúsculas.
Nomenclatura según el tipo de clases	<ul style="list-style-type: none"> • El nombre de las clases controladoras se encontrará seguido de la palabra Controller. Ejemplo de esto la clase: AsignaturasController. • El nombre de los Modelos utilizados dentro de la programación del módulo empieza con letra inicial mayúscula. Ejemplo de esto el modelo Afectaciones. • Clases Export: En las clases Export, después del nombre de la clase le sigue la palabra Export. Ejemplo de esto la clase BalancedecargaExport.
Estándares para comentarios	<ul style="list-style-type: none"> • Los comentarios deben ser oraciones completas. • Si un comentario es una frase u oración, debe comenzar con mayúscula a menos que sea un identificador que comience con minúscula.

	<ul style="list-style-type: none"> • Los comentarios de una línea se encontrarán seguidos de los caracteres “//” en caso de código JavaScript, y deben ubicarse en la misma línea. • Los comentarios de varias líneas para organización del código aparecerán dentro de los caracteres “/** ... */” en caso de que se utilice código JavaScript.
<p>Estándares para espacios en blanco en expresiones y sentencias</p>	<ul style="list-style-type: none"> • No utilizar espacios en blancos en los siguientes escenarios: <ul style="list-style-type: none"> ▪ Dentro de paréntesis, corchetes y llaves. ▪ Antes de una coma, un punto y coma, o dos puntos. ▪ Antes del paréntesis que comienza la lista de argumentos en la llamada a una función. • Deben rodearse con un espacio los siguientes operadores binarios: <ul style="list-style-type: none"> ▪ Asignación (=). ▪ Asignación de aumentación o de resta (+=, -=). ▪ Comparación (==, <, >, >=, <=, !=, <>, in, not in, is, is not). ▪ Expresiones lógicas (and, or, not). • Añadir un espacio después de cada coma “,”.
<p>Otros estándares</p>	<ul style="list-style-type: none"> • Añade un salto de línea antes de una sentencia return (). Ejemplo de esto en el siguiente fragmento de código. <pre data-bbox="786 1304 1484 1503"> public function destroy(Asignaturas \$signatura) { \$signatura->delete(); return redirect()->route('asignaturas.index')->with('info', 'eliminar-asignatura'); } </pre> <ul style="list-style-type: none"> • Utilizar los namespace para todas las clases en los Controller y los Models. Se muestra en los siguientes fragmentos de códigos. <pre data-bbox="833 1661 1500 1728"> namespace App\Http\Controllers\Modulo_Horario; </pre>

```

3 namespace App\Models\Modulo_Horario;

```

3.4 Estrategia de Pruebas

Una estrategia de pruebas es un conjunto de definiciones de alto nivel que determinan cómo se van a llevar a cabo los procesos de pruebas de software, impulsando nuestras decisiones sobre cómo invertir nuestros esfuerzos de pruebas (Averbuj, 2022).

En el presente epígrafe, se pretende mostrar los resultados obtenidos del camino a seguir como estrategia de validación de varias pruebas realizadas a la propuesta de solución creada para el Módulo de Generación de Horarios Docentes para el Sistema de Trabajo Educativo en la Facultad 1, las cuales se pueden observar en la Tabla 3.4, con el objetivo principal de garantizar su calidad.

Tabla 3.4. Estrategias de Pruebas ³MGHDF1.

Tipo de Prueba	Método de Prueba	Validación
Funcional	Casos de prueba (Caja Negra).	Se validan las funcionalidades creadas y diseñadas para el módulo.
Seguridad	Software Acunetix	Valida la seguridad, integridad y confidencialidad básicamente, de los datos en el módulo.
Integración	Integración de módulos	Consiste en la comprobación de que los elementos del software que interactúan entre sí, funcionan de manera correcta.
Carga y Estrés	Software Apache JMeter	Valida el comportamiento principalmente de distintos niveles de usuarios concurrentes y el consumo excesivo de recursos dentro del módulo.

³ MGHDF1: Módulo de Generación de Horarios Docentes para el Sistema de Trabajo Educativo en la Facultad 1.

3.4.1 Pruebas Funcionales

Las pruebas funcionales son técnicas de pruebas que se utiliza para probar las características y funcionalidades principales del sistema o software, en las que se deben cubrir todos los escenarios, incluidas las rutas de falla y los casos límite de las mismas (A, 2016).

Con el objetivo de realizar este tipo de pruebas al Módulo de Generación de Horarios Docentes para el Sistema de Trabajo Educativo de la Facultad 1, se diseñó un conjunto de casos de pruebas, referentes a algunos de los casos de uso del sistema descritos en el capítulo anterior, los que además corresponden a requisitos funcionales de prioridad alta. A continuación, se muestran varios casos de prueba, analizando varios campos de las tablas correspondientes para llevar a cabo dicha prueba, utilizando como valores **V**, para datos válidos, **I**, para datos inválidos, y **N/A**, para datos a los que no es necesario proporcionarles un valor.

Representación de las variables:

Tabla 3.5. Pruebas Funcionales (Insertar Asignaturas).

Variable	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Sesión	Campo de Selección	No	Permite seleccionar la sesión de clases a impartir.
2	Nombre	Campo de texto	No	Permite todos los caracteres.
3	Año Docente	Campo de número	No	Permite el año docente en forma de número entero.
4	Semestre	Campo de número	No	Permite el semestre en forma de número.
5	Estado	Campo de Número	No	Permite el estado de la asignatura (si se encuentra activa o no)

Tabla 3.6. Caso de Prueba 1: RF6_Insertar Asignatura.

Escenario	Descripción	1	2	3	4	5	Respuesta del sistema	Flujo Central
EC 1.1	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar una asignatura, si todos los datos son correctos se agrega la Asignatura en el Sistema.	V	V	V	V	V	Agrega la Asignatura al Sistema y muestra una alerta: Asignatura Registrada.	1. Seleccionar la sección Asignaturas ubicada dentro del Módulo Horario. 2. Seleccionar la opción "Insertar Asignatura". 3. Completar los datos pedidos en el formulario y presionar en el botón "Guardar".
Insertar una Asignatura de manera correcta		Mañana	IP	1	1	1		
EC 1.2	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar una asignatura, si existe algún campo vacío se muestra un mensaje: "Campo Requerido".	V	V	I	V	V	Comprueba que los campos estén vacíos, si existe algún campo vacío, muestra un mensaje que indica que existe algún campo vacío.	
Insertar una Asignatura con campos vacíos.		Mañana	AL		1	1		
EC 1.3	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar una asignatura, si existe alguna asignatura con el mismo nombre registrada se muestra un mensaje: "El campo nombre ya ha sido registrado".	V	V	V	V	V	Comprueba que el nombre de la asignatura que se está insertando no exista ya registrado, si existe, se muestra un mensaje: "El campo nombre ya ha sido registrado".	
Insertar una asignatura con el mismo nombre a otra existente.		Mañana	IP	1	1	1		

Representación de las variables:

Tabla 3.7. Pruebas Funcionales (Insertar Balance de Carga).

Variable	Nombre del Campo	Clasificación	Valor Nulo	Descripción
----------	------------------	---------------	------------	-------------

1	Asignaturas a Impartir	Campo de Selección	No	Permite seleccionar las asignaturas a impartir.
2	Frecuencia de Clases	Campo de número	No	Permite el número de frecuencias de una asignatura por semana.
3	Tipo de Clases	Campo de texto	No	Permite insertar el tipo de clases que se impartirán en una semana de una asignatura.
4	Semana de Clases	Campo de Número	No	Permite insertar el número de la semana en que se dará esas clases.

Tabla 3.8. Caso de Prueba 2: RF17_Insertar Balance de Carga.

Escenario	Descripción	1	2	3	4	Respuesta del sistema	Flujo Central
EC 1.1 Insertar datos en el balance de carga de manera correcta	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar el balance de carga, si todos los datos son correctos se insertan los datos en el Sistema.	V AL	V 2	V C, CP	V 1	Agrega los datos del balance de carga al Sistema y muestra una alerta: "Datos Registrados".	1. Seleccionar la sección Balance de Carga ubicada dentro del Módulo Horario. 2. Seleccionar la opción "Insertar Datos". 3. Completar los datos pedidos en el formulario y presionar en el botón "Guardar".

EC 1.2 Insertar datos en el balance de carga de con campos vacíos.	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar el balance de carga, si existe algún campo vacío se muestra un mensaje: "Campo Requerido".	V	V	V	I	Comprueba que los campos estén vacíos, si existe alguno, muestra un mensaje que indica que existe algún campo vacío.
		IP	2	CP, C		

Representación de las variables:

Tabla 3.9. Pruebas Funcionales (Insertar Afectaciones).

Variable	Nombre del Campo	Clasificación	Valor Nulo	Descripción
1	Profesor Afectado	Campo de Selección	No	Permite seleccionar en caso que sea necesario un profesor que estará afectado un turno de clase específico.
2	Semana	Campo de número	No	Permite insertar la semana en forma de número entero.
3	Día de la Semana Afectado	Campo de Selección	No	Permite seleccionar el día que se afectará.
4	Turno de Clases	Campo de Número	Sí	Permite insertar el turno de clases que

				se afectará en específico.
5	Año Docente	Campo de Número	No	Permite el número del año docente que se afectará.

Tabla 3.10. Caso de Prueba 3: RF11_Insertar Afectaciones.

Escenario	Descripción	1	2	3	4	5	Respuesta del sistema	Flujo Central
EC 1.1	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar una afectación, si todos los datos son correctos se agrega la Afectación en el Sistema.	V	V	V	V	V	Agrega la Afectación al Sistema y muestra una alerta: Afectación Registrada.	1. Seleccionar la sección Afectaciones ubicada dentro del Módulo Horario. 2. Seleccionar la opción "Insertar Afectación".
Insertar una Afectación de manera correcta.		Richard Castañet Blanco	1	Lunes	1	1		
EC 1.2	Interfaz con el formulario correspondiente para llenar los datos necesarios para insertar una afectación, si existe algún campo vacío se muestra un	I	I	V	V	V	Comprueba que los campos estén vacíos, si existe alguno, muestra un mensaje que indica que existe algún campo vacío.	3. Completar los datos pedidos en el formulario y presionar en el botón "Guardar".
Insertar una Afectación con campos vacíos.				Lunes	1	1		

	mensaje: "Campo Requerido".							
--	--------------------------------	--	--	--	--	--	--	--

Las pruebas funcionales se realizaron en 3 iteraciones, que permitieron conocer la calidad del módulo que se está construyendo. Como resultado final se obtuvo un total de 10 no conformidades en la primera iteración, donde se encuentra cinco de ortografía, tres de redacción, y dos de validación, de las cuales se resolvieron tres de ortografías. Durante el desarrollo de la segunda iteración de pruebas se corrigieron las 2 no conformidades pendientes en la primera iteración relacionadas con la ortografía, así como las 3 no conformidades relacionadas a la redacción, y, además, se detectaron dos nuevas no conformidades de funcionalidad, de manera que solo se mantuvo pendiente las dos no conformidades relacionadas con las validaciones, y las dos no conformidades nuevas relacionadas a las funcionalidades. Durante el desarrollo de la tercera iteración no se encontraron no conformidades nuevas, y se solucionaron los cuatros no conformidades pendientes de la segunda iteración. Todos los resultados anteriormente descritos se pueden observar en el siguiente gráfico:

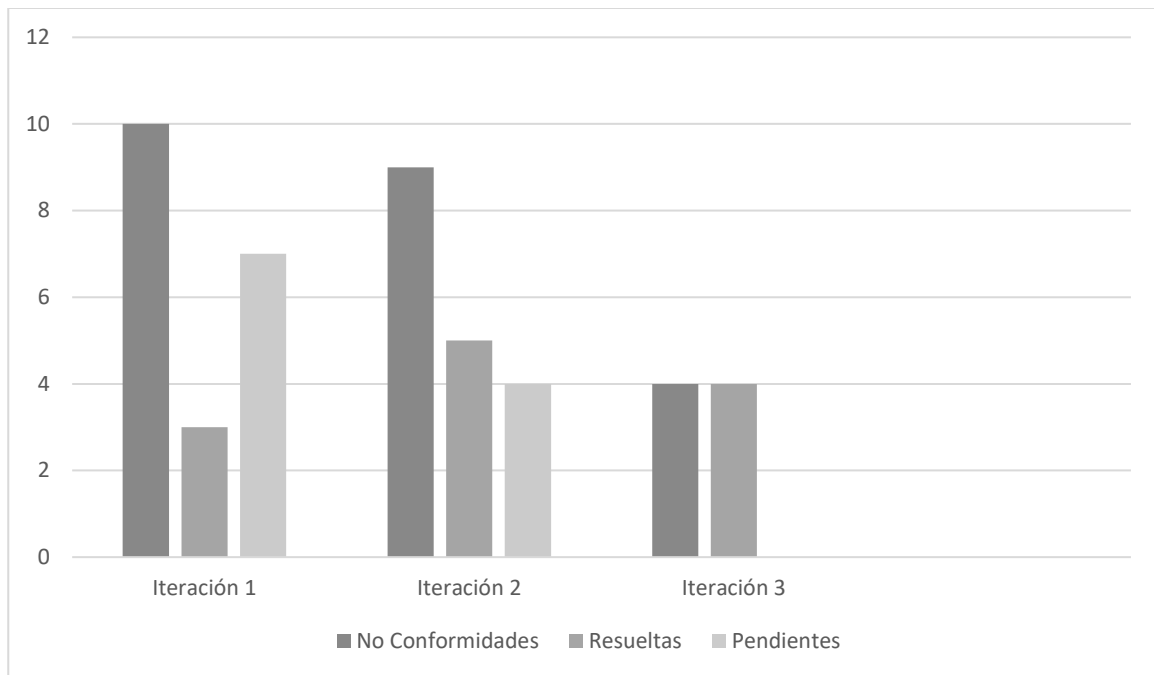


Figura 19. Pruebas de Funcionalidad.

Las no conformidades de validación encontradas, se resolvieron mediante el uso de JavaScript, de manera que se validaron dos formularios existentes en el módulo correspondientes a: Generar Horario, y Buscar Horario respectivamente, que permitían enviar el formulario con algunos campos en blancos. Las no conformidades de funcionalidad encontradas estaban relacionadas a las funcionalidades de Editar Asignatura, y Editar Balance de Carga, estas se solucionaron realizando nuevamente la programación necesaria en cada uno de los métodos, de manera que se editaran las asignaturas y los balances de carga de manera adecuada.

Una vez culminadas las pruebas funcionales realizadas al módulo desarrollado, se puede arribar a la conclusión de que el mismo se encuentra en óptimas condiciones funcionales para su uso futuro uso.

3.4.2 Pruebas de Seguridad

Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en las aplicaciones web, de manera que buscan disminuir el impacto de ataques a ellas y pérdidas de información importante (Diaz, 2014).

Con el objetivo de realizar una exitosa prueba de seguridad de la propuesta de solución desarrollada, se tuvo en cuenta la herramienta Acunetix Web Vulnerability Scanner, herramienta la cual fue descrita con anterioridad en el epígrafe 1.3. Durante la primera iteración realizada, la aplicación detectó un total de 18 vulnerabilidades, divididas en tres secciones principalmente: de nivel medio doce vulnerabilidades, de nivel bajo tres vulnerabilidades y de carácter informal tres vulnerabilidades. De las vulnerabilidades de nivel medio, la minoría de ellas son debido al uso de protocolos no seguros para el envío de datos, y el resto se deben al envío de mensajes de mensajes de error utilizados por Laravel. Las vulnerabilidades de nivel bajo, están relacionadas generalmente a los posibles ataques a la página de autenticación y las posibles redirecciones que pueden surgir a partir de la misma una vez que el usuario está dentro del sistema. De tipo Informativo se obtuvo varias vulnerabilidades, relacionados a los campos correspondientes a las contraseñas y los campos repetir contraseña. En la Figura 20. Herramienta Acunetix, se pueden apreciar los resultados obtenidos de las pruebas.

Alerts distribution

Total alerts found	18
High	0
Medium	12
Low	3
Informational	3

Figura 20. Herramienta Acunetix

Luego de analizados las vulnerabilidades anteriormente mencionadas se puede concluir que las mismas, son debido la mayoría, a mensajes de error y de redirección propios del Framework Laravel utilizado en la implementación, los cuales se utilizan para distintas funcionalidades dentro del sistema. En la segunda iteración de la aplicación no se obtuvo ninguna vulnerabilidad nueva en la seguridad del sistema, permitiendo esto poder utilizar la herramienta desarrollada como propuesta de solución de manera segura. A continuación, en la siguiente figura se pueden observar todos los resultados antes descritos:

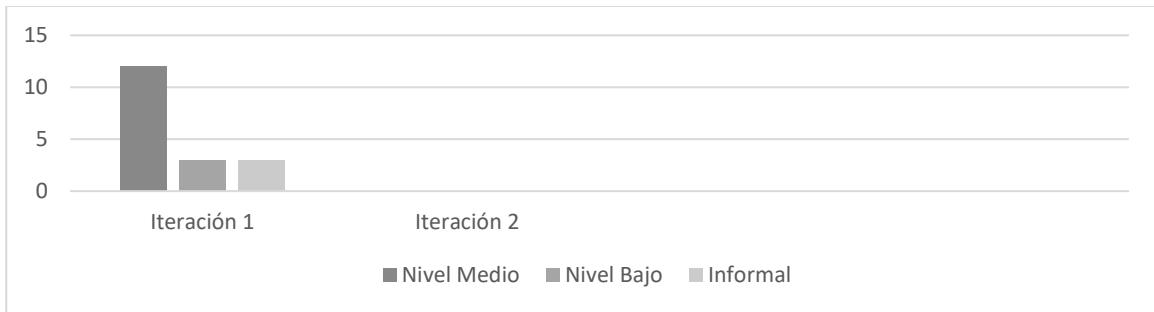


Figura 21. Pruebas de Seguridad

Una vez concluidas las iteraciones correspondientes a las pruebas de seguridad, se puede asegurar que tanto el Sistema de Trabajo Educativo de la Facultad 1 como el módulo desarrollado cumple con los requisitos necesarios para garantizar una buena seguridad del mismo.

3.4.3 Pruebas de Integración

Las pruebas de integración comprueban las funciones externas tras las pruebas de componentes, además comprueba la interacción entre los elementos de software entre distintos sistemas o hardware y software (Díaz, 2020).

En el caso de la solución desarrollada, para la verificación de una correcta interoperabilidad entre los módulos desarrollados para el Sistema de Trabajo Educativo de la Facultad 1, se llevaron a cabo las acciones siguientes:

- Integración del Módulo de generación de horarios docentes al Sistema de Trabajo Educativo en la Facultad 1.

Se muestra, en la Figura 22. Pruebas de Integración, los resultados de la ejecución de las dos iteraciones de las pruebas de integración realizadas al sistema:

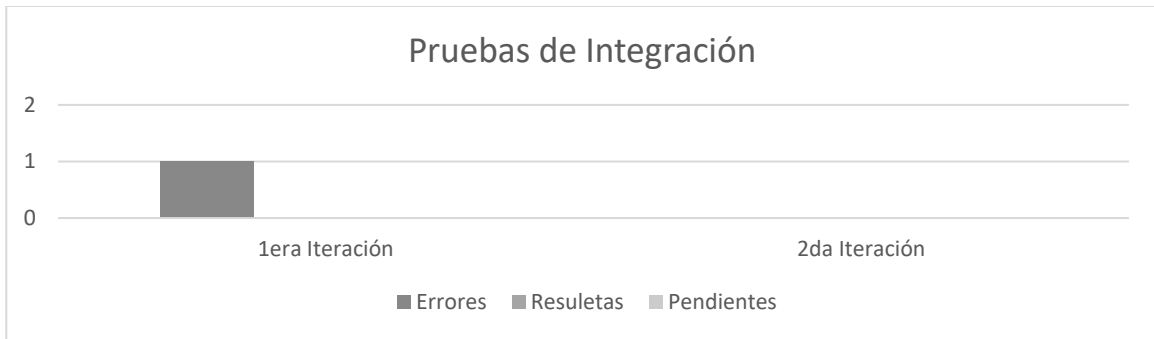


Figura 22. Pruebas de Integración

Durante el proceso de ejecución de las pruebas de integración se pudo comprobar de manera conjunta cada uno de los componentes del Sistema de Trabajo Educativo de la Facultad 1. Se realizó un total de 2 iteraciones durante las pruebas de integración, durante la primera iteración se obtuvo un error debido a que no existía relación entre el modelo de datos del sistema y el modelo de datos del módulo que se pretendía integrar, dicho error se corrigió la misma iteración. Durante la segunda iteración no se encontró ningún error nuevo.

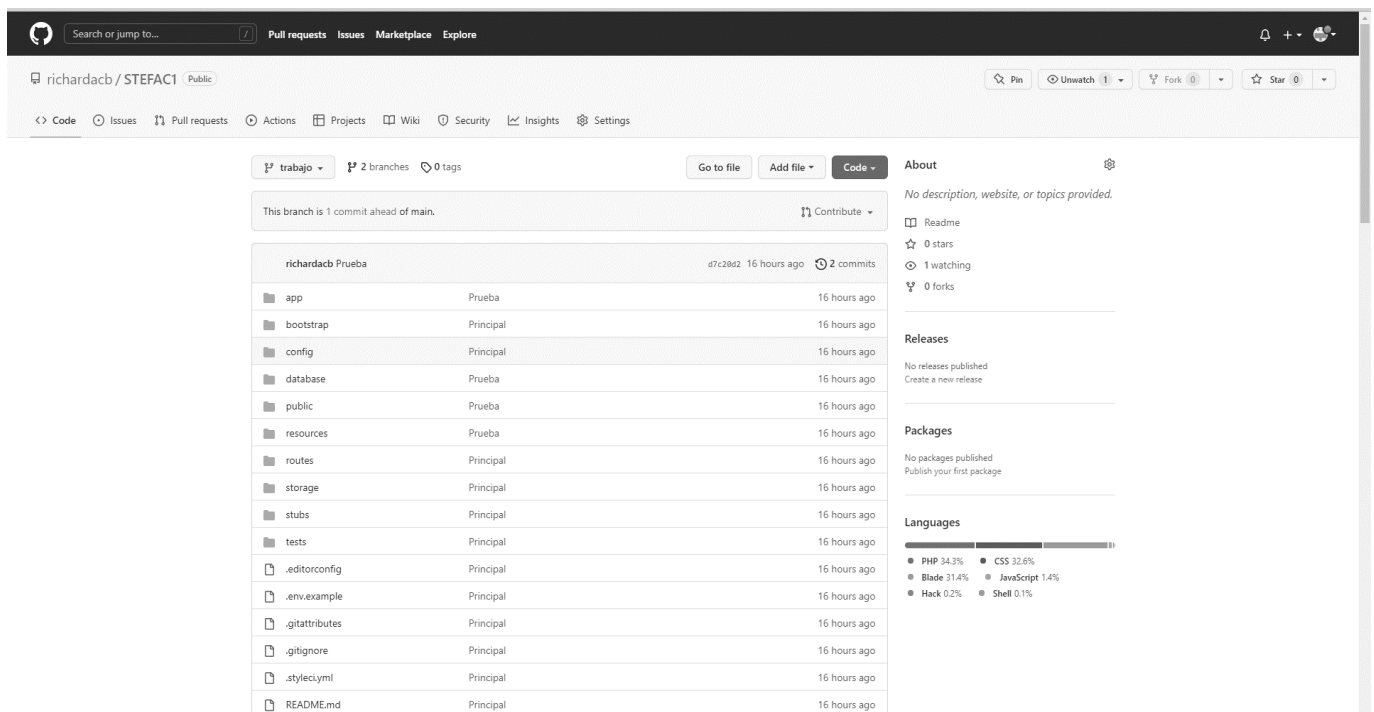


Figura 23. Pruebas de Integración-Github

Con las pruebas de integración realizadas se obtuvo un Sistema de Trabajo Educativo para la Facultad 1, en el cual ya se encuentran en funcionamiento todos los módulos que requiere dicho sistema.

3.4.4 Pruebas de Carga y estrés

La prueba de carga y estrés son pruebas que se utilizan para determinar los límites del sistema, donde el objetivo es verificar la estabilidad y fiabilidad del sistema en condiciones extremas. Para la realización de esta prueba se utilizó la herramienta Apache JMeter, descrita en el epígrafe 1.3. Las pruebas se realizaron desde un ordenador con 8GB de RAM, microprocesador Intel Core i5 7th Gen, con 2.60 GHz y sistema operativo Windows 10. A continuación, se describen las variables que miden el resultado de las pruebas de carga y estrés realizadas al módulo:

Muestra: Cantidad de peticiones realizadas para cada URL.

Media: Tiempo promedio en milisegundos en el que se obtienen los resultados.

Mediana: Tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.

Min: Tiempo mínimo que demora un hilo en acceder a una página.

Max: Tiempo máximo que demora un hilo en acceder a una página.

Línea 90 %: Máximo tiempo utilizado por el 90 % de la muestra, al resto de la misma le llevo más tiempo.

% Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.

Rendimiento (Rend): El rendimiento se mide en cantidad de solicitudes por segundo.

Kb/s: Velocidad de carga de las páginas.

Como se muestra en la siguiente tabla, se simularon las peticiones realizadas al módulo por un total de 50, 100 y 150 usuarios simultáneamente en cada caso. Se obtuvieron los siguientes resultados:

Tabla 3.11. Pruebas de Cargas y Estrés

Usuarios	Muestras	Media	Mediana	Min	Max	Línea 90%	%Error	Rend	Kb/s
50	132	16503	16512.35	65	48466	2902	0.02%	5	6276.7
100	163	22779	18542.44	69	48763	2591	0.50%	10	5717.6
150	190	24769	20745.30	73	48935	2193	1.25%	15	5434.7

Las pruebas realizadas muestran que el Sistema de Trabajo Educativo de la Facultad 1 es capaz de responder a 132 peticiones de 50 usuarios conectados simultáneamente en un tiempo promedio de 16503 milisegundos (1.6 segundos aproximadamente) con 0.02 % de error, esto evidencia que el sistema puede procesar la carga esperada.

Por otra parte, se realizaron 163 peticiones iniciadas por 100 usuarios y en este caso el sistema respondió en 22779 milisegundos (2.2 segundos aproximadamente) como tiempo promedio. Esto demuestra que el sistema puede procesar la carga esperada, aunque no fue capaz de responder correctamente el 0.50% de las peticiones realizadas. Por último, y con el objetivo de analizar el comportamiento del sistema en condiciones extremas, se realizó una prueba de estrés para un conjunto de 150 usuarios conectados simultáneamente. En este caso, el sistema responde a las 190 peticiones en un tiempo promedio de 24769 milisegundos (2.4 segundos aproximadamente), pero con un porcentaje de error de 1.25%. Este resultado está estrechamente relacionado al entorno donde se realizó la prueba, el cual no es un servidor dedicado sino un cliente habilitado para realizar la prueba.

Una vez finalizadas las pruebas de Carga y Estrés realizadas al Sistema de Trabajo Educativo de la Facultad 1, se pueden concluir que el sistema actualmente soporta la interacción simultánea de varios usuarios al mismo tiempo, lo que se considera como un punto positivo que permitirá el correcto uso del sistema.

3.5 Resultados del MGHDF1

En las siguientes imágenes se puede visualizar el resultado obtenido una vez llevada a cabo toda la implementación de la propuesta de solución:

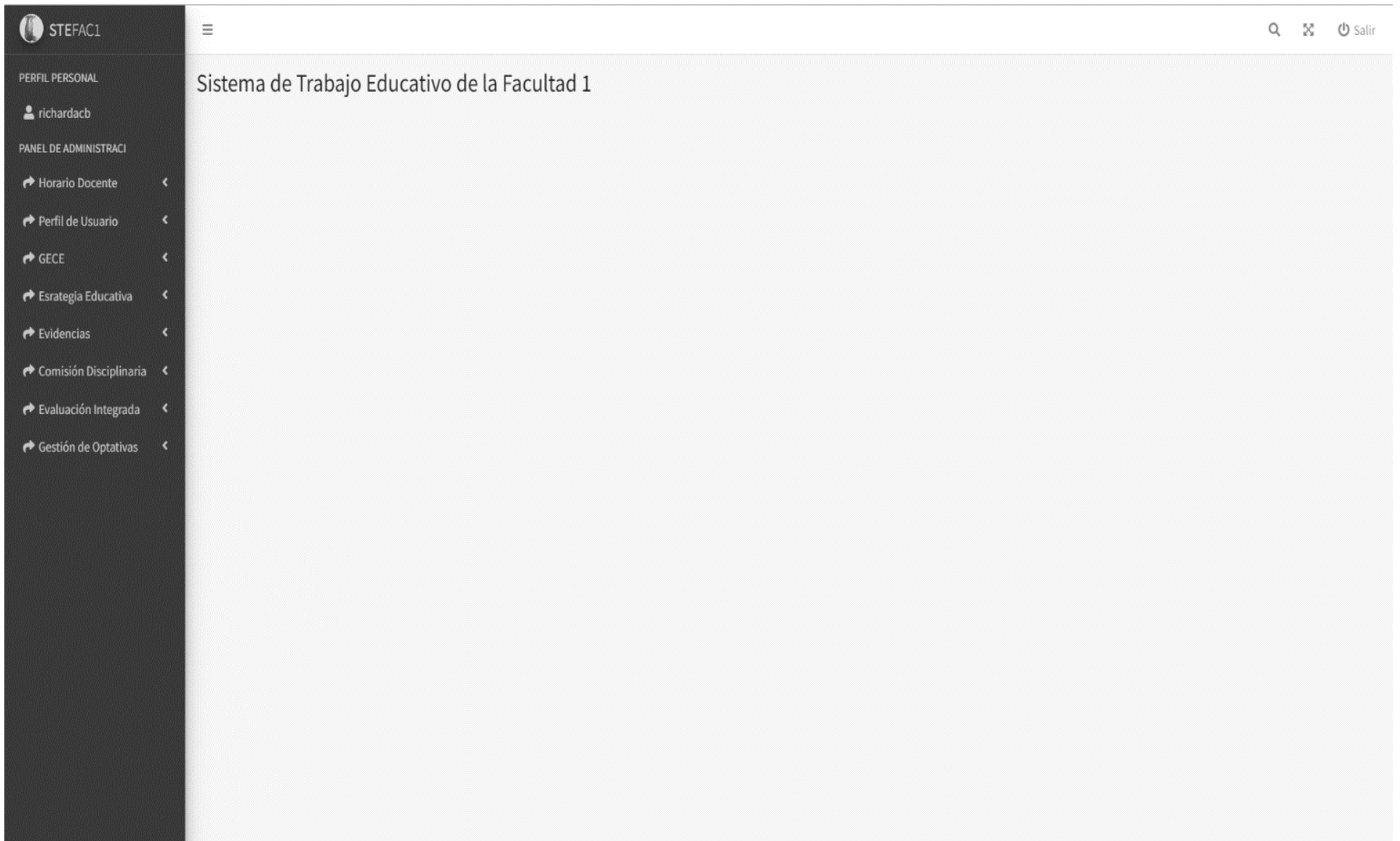


Figura 24. Portada Inicial del Sistema de Trabajo Educativo de la Facultad 1.

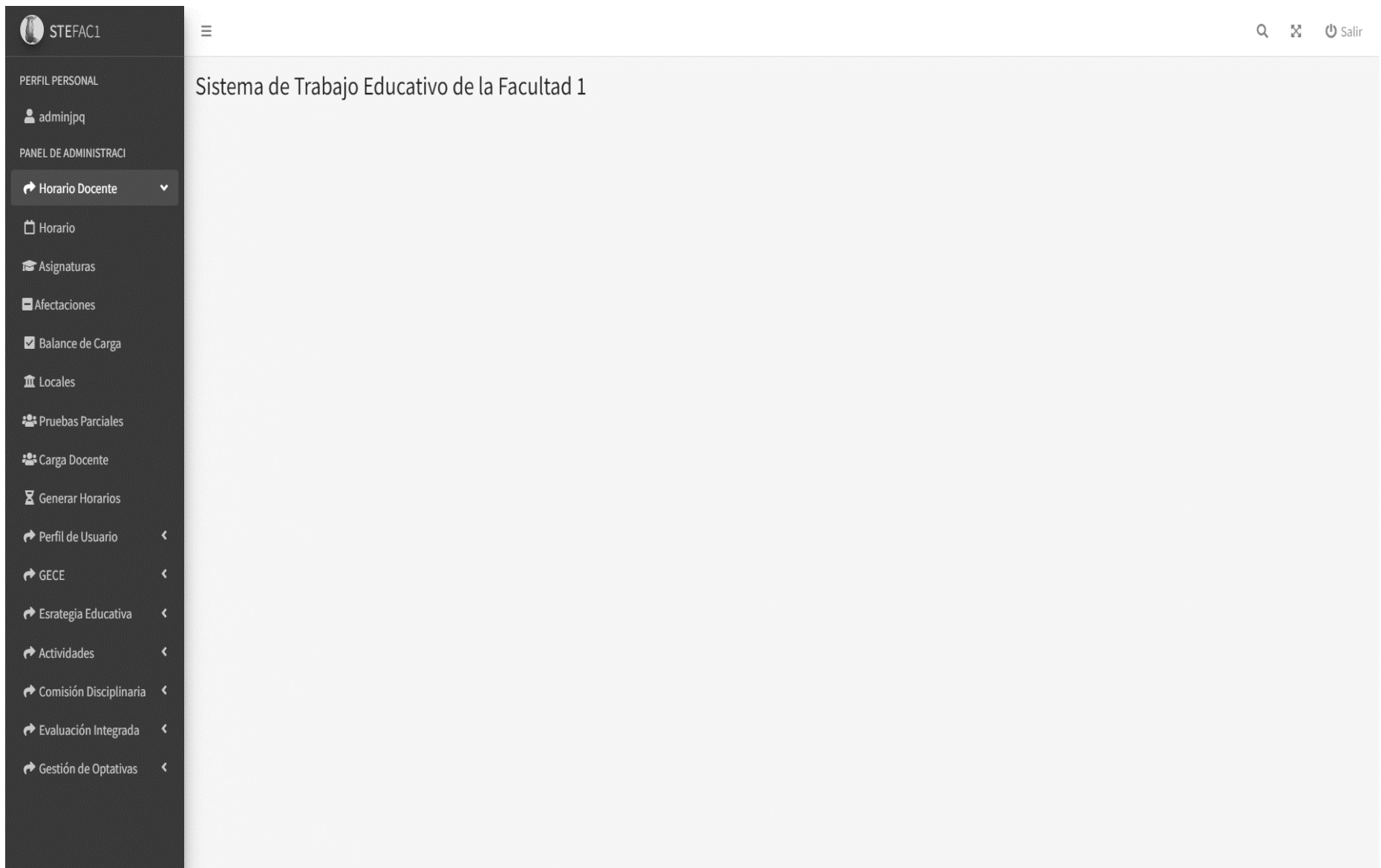


Figura 25. Módulo Horario.

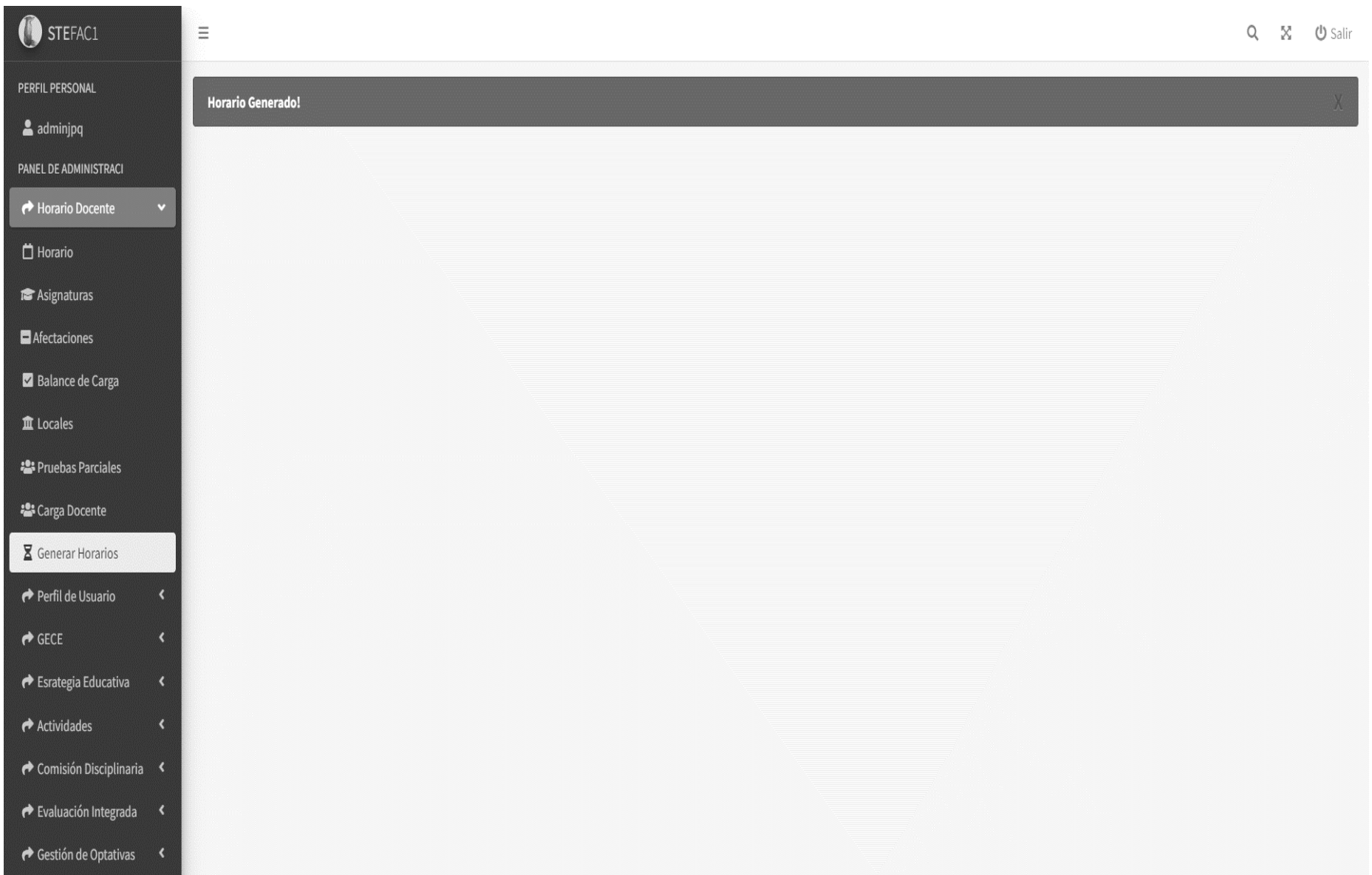


Figura 26. Vista Generar-Horario

STEFAC1

PERFIL PERSONAL

adminjq

PANEL DE ADMINISTRACION

Horario Docente

Horario

Asignaturas

Afectaciones

Balance de Carga

Locales

Pruebas Parciales

Carga Docente

Generar Horarios

Perfil de Usuario

GECE

Estrategia Educativa

Actividades

Comisión Disciplinaria

Evaluación Integrada

Gestión de Optativas

Salir

Turno	Lunes	Martes	Miercoles	Jueves	Viernes
1		AL(C)Salon-401			
2					
3					
4					
5					
6					

Año Docente

1

Semana del Curso

1

Grupo

IDF1101

Buscar

Figura 27. Visualizar Horario

3.6 Validación Científica

Con el objetivo de lograr una validación científica de la presente investigación se utiliza la técnica de ladov, descrita durante la introducción de la investigación. Para aplicar dicha técnica se realizó una encuesta a los Profesores Principales de Año de la Facultad 1 de la Universidad de las Ciencias Informáticas, la misma se encuentra estructurada en un total de 5 preguntas, donde tres son preguntas cerradas y dos abiertas, las cuales se pueden observar en el Anexo 3 del presente documento. A continuación, se muestra la Tabla 3.12, donde se relacionan las 3 preguntas cerradas.

Tabla 3.12. Cuadro lógico de ladov

3- Una vez observado el proceso de generación del Horario Docente y visualizado sus resultados, sería tan amable de calificar el módulo con una de las siguientes opciones:	1- ¿Considera necesario un Módulo para la Generación de Horarios Docentes para el Sistema de Trabajo Educativo de la Facultad 1?								
	Sí			No Sé			No		
	2- ¿Considera útil el Módulo de Generación de Horarios Docentes para el Sistema de Trabajo Educativo para la Facultad 1?								
	Sí	No	No Sé	Sí	No	No Sé	Sí	No	No Sé
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	3	4
No me gusta nada	6	6	6	6	4	4	3	3	4
No sé decir	2	3	6	3	3	3	6	6	4

Esta técnica además permite obtener el índice de satisfacción grupal (ISG), para lo cual se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma: Máxima satisfacción (+ 1), Satisfecho (0.5), No definido (0), Insatisfecho (- 0.5) y Máxima insatisfacción (- 1). A partir de la cantidad de respuestas obtenidas en la encuesta es posible calcular el ISG siguiendo la siguiente fórmula:

$$ISG = \frac{A(+1) + B(0.5) + C(0) + D(-0,5) + E(-1)}{N}$$

En la anterior fórmula, A, B, C, D y E representan la cantidad de personas que interactúan con la aplicación y N representa el total de personas. La escala de valores del índice grupal que se toma al aplicar la técnica de ladov se encuentra organizada de la siguiente manera:

- -1 a -0.5 = Insatisfacción.
- -0.49 a 0.49 = Contradicción.
- 0.5 a 1 = Satisfacción.

Luego de aplicada la encuesta, los resultados obtenidos llevados a la escala de satisfacción fueron los siguientes:

Tabla 3.13. Resultados Obtenidos de la Encuesta

Usuarios(N)	5	Escala
Máxima satisfacción	3	A
Satisfecho	1	B
No definido	1	C
Insatisfecho	0	D
Máxima insatisfacción	0	E

A continuación, se realiza el cálculo del Índice de Satisfacción Grupal:

$$ISG = \frac{3(+1) + 1(0.5) + 1(0) + 0(-0.5) + 0(-1)}{5} = 0.7$$

A raíz del análisis de la encuesta llevada a cabo a los Profesores Principales de Año, se puede concluir que el uso de la técnica de ladov permitió conocer el nivel de aceptación de la propuesta de solución, reflejado en el alto índice de satisfacción grupal obtenido (ISG = 0.7). Además, las respuestas obtenidas por los encuestados respecto a las preguntas abiertas, reafirman las ventajas que traerá consigo la puesta en marcha del Módulo de Generación de Horario Docente para la Facultad 1, por lo que se puede afirmar que la propuesta de solución satisface el objetivo principal de la investigación.

CONCLUSIONES PARCIALES DEL CAPÍTULO

En el capítulo luego de abordados todo lo relacionado a la implementación y pruebas necesarias al módulo a desarrollar, se puede concluir que la elaboración de los diagramas de componentes permite adquirir una mejor comprensión de la estructura del módulo a implementar. Además, el adecuado uso de los estándares de codificación permite lograr que el código fuente del módulo desarrollado posea una estructura legible y

de fácil comprensión, lo que a su vez conlleva a la obtención de una aplicación funcional y completamente operativa. Por otra parte, la estrategia de pruebas construida en el capítulo, junto al desarrollo y validación de las mismas, unido a la prueba de validación científica, han permitido obtener resultados positivos acerca del módulo implementado, permitiendo así que el mismo cumpla con los requisitos definidos junto al cliente.

CONCLUSIONES GENERALES

De manera general la presente investigación concluyó con la creación del Módulo para la Generación de Horarios Docentes integrado al Sistema de Trabajo Educativo de la Facultad 1, sirviendo el mismo de apoyo al proceso docente-educativo en la Facultad 1. Entre los principales aspectos significativos que se pueden recalcar del Módulo se encuentran:

1. El estudio del arte realizado contribuyó a la recopilación de información y tendencias acerca de los sistemas y módulos de planificación y confección de horarios docentes, lo que permitió obtener una mayor comprensión del alcance de la investigación.
2. El análisis de las herramientas y las tecnologías han permitido definir y abordar sobre el desarrollo de la propuesta de solución para el módulo de generación horarios docentes, permitiendo así la definición del entorno de desarrollo adecuado para la implementación.
3. La unión de varias áreas relacionadas con la programación y la ingeniería de software, además de base de datos, entre otras, permitió llevar a cabo el correcto análisis, diseño e implementación del Módulo de Generación de Horarios Docentes de la Facultad 1.
4. La validación de las estrategias de pruebas definidas, logró probar el correcto funcionamiento del Módulo de Generación del Horario Docente de la Facultad 1, teniendo en cuenta para esto los requisitos definidos junto al cliente. Además, con la validación científica llevada a cabo mediante el uso de la técnica de ladov, permitió concluir que el módulo desarrollado cumple con el objetivo principal de la investigación.

RECOMENDACIONES

Luego de realizada la técnica de ladov se obtuvieron varias recomendaciones, de las cuales el autor propone para el desarrollo de futuras investigaciones relacionadas al tema:

- 1- Implementar una opción de “Editar Horario Docente”, una vez que ya esté generado el mismo.

REFERENCIAS BIBLIOGRÁFICAS

- A, J. (2016, febrero 1). *Functional Testing: A Complete Guide with Types and Example*. Software Testing Help.
<https://www.softwaretestinghelp.com/guide-to-functional-testing/>
- Abreu Guerra, L. (2012). *Sistema de Gestión para la planificación docente en la Facultad Regional Mártires de Artemisa*. https://repositorio.uci.cu/jspui/handle/ident/TD_06167_12
- Álvarez, M. A., Rousset, D., & Peña Tresancos, J. (2017). *Manual de CSS3*. mardeasa.es.
- Apache JMeter*. (2022). <https://jmeter.apache.org/>
- artanaisi. (2019, enero 21). *Gaceta Oficial No. 25 Ordinaria de 2018* [Text]. Gaceta Oficial.
<https://www.gacetaoficial.gob.cu/es/gaceta-oficial-no-25-ordinaria-de-2018>
- Averbuj, A. (2022, marzo 7). Cómo crear la Estrategia de Pruebas adecuada para tu proyecto | Abstracta Chile. *Blog de Testing y Calidad de Software | Abstracta Chile*. <https://cl.abstracta.us/blog/guia-crear-estrategia-pruebas-software-adecuada/>
- baltolkien. (2014, julio 22). FET, el generador de horarios libre. *KDE Blog*. <https://www.kdeblog.com/fet-el-generador-de-horarios-libre.html>
- Baratutes González, E. (2018). *Sistema para la planificación y control de inventarios en el proceso de personalización de documentos de identificación* [BachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1].
<https://repositorio.uci.cu/jspui/handle/123456789/9927>
- Carrizo, D., Rojas, J., Carrizo, D., & Rojas, J. (2018). Metodologías, técnicas y herramientas en ingeniería de requisitos: Un mapeo sistemático. *Ingeniare. Revista chilena de ingeniería*, 26(3), 473-485.
<https://doi.org/10.4067/S0718-33052018000300473>
- Chaparro, R., & Vicente, C. (2020). *Diagramas esenciales del lenguaje unificado de modelado para los requisitos agiles en el desarrollo de software*. <http://repository.unad.edu.co/handle/10596/38052>

- Comes, A. (2022). *Programa Generador de Horarios Escolares*. Custombit. <https://www.custombit.com.ar/generar-horarios-docentes/index.html>
- contributors, M. O., Jacob Thornton, and Bootstrap. (s. f.). *Bootstrap*. Recuperado 17 de mayo de 2022, de <https://getbootstrap.com/>
- Costa, G. I. (2007). Akademos, un Sistema Automatizado para la Gestión Académica. *Serie Científica de la Universidad de las Ciencias Informáticas*, 1(1), Article 1. <https://publicaciones.uci.cu/index.php/serie/article/view/253>
- Daudinot Hamiltón, Y. (2015a). *Perfil de usuario para el Sistema de Planificación de Actividades SIPAC*. <https://repositorio.uci.cu/jspui/handle/123456789/7304>
- Daudinot Hamiltón, Y. (2015b). *Perfil de Usuario para el Sistema de Planificación de Actividades SIPAC*. Universidad de las Ciencias Informáticas.
- Desarrollo Web. (2022). *El diagrama de casos de uso en UML*. IONOS Digital Guide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>
- Díaz, P., Moreno, S., & Aedo, I. (2005). *Ingeniería de la web y patrones de diseño*.
- Diaz, S. M. (2014). *Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software*.
- Documentation for Visual Studio Code*. (s. f.). Recuperado 17 de mayo de 2022, de <https://code.visualstudio.com/docs>
- Educaria. (2022). Horarios. *Educaria*. <https://www.educaria.com/horarios/>
- Fernández de Castro, A. (2020). El proceso de validación mediante la técnica de ladov en cursos por encuentros. 2020.

- Flores, J. O. C. (2015). Concepto de planeación. *Boletín Científico de las Ciencias Económico Administrativas del ICEA*, 3(6), Article 6. <https://doi.org/10.29057/icea.v3i6.166>
- García Rojas, I. D. (2015). *Sistema de gestión para la planificación y control del horario docente en la Facultad 3* [Tesis]. Universidad de las Ciencias Informáticas.
- González Herrera, C. Y. (2017). *Módulo para el diseño de modelos entidad relación en la Plataforma RDB-Learning*. Universidad de las Ciencias Informáticas.
- González Flores, I., Cardoza Rodríguez, H., Bullaín Diéguez, R. R., & Medrano Broche, B. E. (2010). *Implementación de una Biblioteca de Algoritmos para la generación automática de horarios de la aplicación web Ghordo*. https://repositorio.uci.cu/jspui/handle/ident/TD_03420_10
- González, Y. D., & Romero, Y. F. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, 11(1), 47-57.
- Introducción | CSS 3*. (s. f.). Recuperado 17 de mayo de 2022, de <https://www.arkaitzgarro.com/css3/capitulo-1.html>
- JavaScript. (2022). *¿Qué es JavaScript? - Aprende sobre desarrollo web | MDN*. https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- Laravel. (2022). *Laravel—The PHP Framework For Web Artisans*. <https://laravel.com/>
- León, Y. (2022). <https://rc.upr.edu.cu/handle/DICT/3511>
- Luna, F. (2019). *JavaScript—Aprende a programar en el lenguaje de la web*. RedUsers.
- Manual de CSS 3*. (s. f.). DesarrolloWeb.Com. Recuperado 17 de mayo de 2022, de <https://desarrolloweb.com/manuales/css3.html>
- Marín Díaz, A., Trujillo Casañola, Y., Buedo Hidalgo, D., Marín Díaz, A., Trujillo Casañola, Y., & Buedo Hidalgo, D. (2020). Estrategia de pruebas para organizaciones desarrolladoras de software. *Revista Cubana de Ciencias Informáticas*, 14(3), 83-104.

- Ortíz Ramos, M., & Torres Iglesias, Y. (2008). *Análisis y Diseño de un sistema para la planificación automatizada del Horario Docente de la facultad 4*. https://repositorio.uci.cu/jspui/handle/ident/TD_1488_08
- Peñalara. (2022). *GHC timetables generator for educational centres | Peñalara GHC*.
<https://www.penalara.com/en/ES/whatisghc/>
- PHP. (s. f.). *PHP: What can PHP do? - Manual*. Recuperado 17 de mayo de 2022, de
<https://www.php.net/manual/en/intro-whatcando.php>
- Qué es HTML5: Definición y funcionamiento*. (2019, enero 20). OpenWebinars.net.
<https://openwebinars.net/blog/que-es-html5/>
- Rodríguez, L. (2017). *Guía para diseñar programas de diplomado por competencias profesionales integradas. 2017*.
- Rodríguez, M. R., & López, S. T. (2019). CMMI, un modelo de procesos que norma la calidad. *Serie Científica de la Universidad de las Ciencias Informáticas*, 12(10), 117-125.
- Rodríguez Sánchez, T. (2015). *Metodología de desarrollo para la actividad productiva de la UCI*.
- Romero Romero, L., & Nasco Castrillón, D. (2019). *Sistema para la gestión del proceso de trabajos de diplomas en la Facultad 3*. <https://repositorio.uci.cu/jspui/handle/123456789/9988>
- Sánchez, M. I. (2011). *Desarrollo Orientado a Objetos con UML*.
<https://www.uv.mx/personal/maymendez/files/2011/05/umltotal.pdf>
- SHAHU GAIKWAD, S. (2019). A review Paper on Bootstrap Framework. *April 2019*.
- SOMMERVILLE, I. (2005). *Ingeniería del Software* (Séptima Edición). PEARSON EDUCACIÓN. S.A.
- Sparks, G. (2022). Una Introducción al UML. En *El Modelo Lógico*. Enterprise Architect.
- Spurlock, J. (2013). *Bootstrap: Responsive Web Development*. O'Reilly Media, Inc.

Tabarés Gutiérrez, R. (2016). El surgimiento de HTML5; un nuevo paradigma en los estándares Web. *Teknokultura*.

Revista de Cultura Digital y Movimientos Sociales. https://doi.org/10.5209/rev_TK.2016.v13.n1.52152

Visual Paradigm. (2022). *Visual Paradigm—UML, Agile, PMBOK, TOGAF, BPMN and More!* <https://www.visual-paradigm.com/features/>

Vulnerability Scanner. (2022). Acunetix. <https://www.acunetix.com/vulnerability-scanner/>

What are the uses of JavaScript—Javatpoint. (s. f.). [Www.Javatpoint.Com](http://www.javatpoint.com). Recuperado 17 de mayo de 2022, de <https://www.javatpoint.com/what-are-the-uses-of-javascript>

Why Visual Studio Code? (s. f.). Recuperado 17 de mayo de 2022, de <https://code.visualstudio.com/docs/editor/whyvscode>

XAMPP. (2022). <https://www.apachefriends.org/es/index.html>

ANEXOS

Anexo 1. Encuesta realizada al personal encargado de la planificación y confección del Horario Docente de la Facultad 1.

- 1- ¿Cómo se realiza actualmente el proceso de planificación y confección del horario docente de la Facultad 1?
- 2- ¿Quiénes son los actores que participan en el proceso de planificación y confección del horario docente?
- 3- ¿Qué nivel de complejidad presenta el proceso de planificación y confección del horario docente?

Anexo 2. Encuesta realizada al personal encargado de la planificación y confección del Horario Docente de la Facultad 1, con el objetivo de realizar la captura de los requisitos funcionales.

- 1- ¿Qué información se necesita recopilar de las asignaturas de cada año docente?
- 2- ¿Quiénes son las personas de la Facultad autorizadas a generar el Horario Docente?
- 3- ¿Cuáles son las inconvenientes que pueden surgir durante la planificación del Horario Docente?
- 4- ¿Cuáles son los datos específicos que se necesitan para la generación automática del Horario Docente de cada año?

Anexo 3. Encuesta realizada a los profesores principales de año de la Facultad 1.

Profesores Jefes de Año:

- 1- ¿Considera necesario un Módulo para la Generación de Horarios Docentes para el sistema de Trabajo Educativo de la Facultad 1?
Sí___ No___ No Sé___
- 2- ¿Considera útil el Módulo de Generación de Horarios Docentes para el Sistema de Trabajo Educativo para la Facultad 1?
Sí___ No___ No Sé___

3- Una vez observado el proceso de generación del Horario Docente y visualizado sus resultados, sería tan amable de calificar el módulo con una de las siguientes opciones:

Me gusta mucho Me disgusta más de lo que me gusta

Me gusta más de lo que me disgusta No me gusta nada

Me da lo mismo No sé decir

4- ¿Qué importancia le otorga a la creación del módulo de generación de horarios docentes para el sistema de trabajo educativo de la Facultad 1?

5- ¿Consideran ustedes que el Módulo de Generación de Horario Docente del Sistema de Trabajo Educativo de la Facultad 1 pueda ser utilizado por otras Facultades de la Universidad?